



TUGAS AKHIR - KS141501

**PENERAPAN METODE KOMBINASI ALGORITMA GENETIKA
DAN TABU SEARCH DALAM OPTIMASI ALOKASI KAPAL
PETI KEMAS (STUDI KASUS : PT. XYZ)**

***APPLICATION OF A HYBRID GENETIC ALGORITHM AND
TABU SEARCH METHOD FOR CONTAINER SHIPS
ALLOCATION OPTIMIZATION (CASE STUDY: PT. XYZ)***

**NOVIAN TIANDINI
NRP 5213 100 075**

**Dosen Pembimbing
Wiwik Anggraeni S.Si, M.Kom**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

TUGAS AKHIR - KS 141501

**PENERAPAN METODE KOMBINASI ALGORITMA
GENETIKA DAN TABU SEARCH DALAM OPTIMASI
ALOKASI KAPAL PETI KEMAS (STUDI KASUS : PT.
XYZ)**

**NOVIAN TIANDINI
NRP 5213 100 075**

**Dosen Pembimbing
Wiwik Anggraeni S.Si, M.Kom**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

FINAL PROJECT - KS 141501

***APPLICATION OF A HYBRID GENETIC ALGORITHM
AND TABU SEARCH METHOD FOR CONTAINER SHIPS
ALLOCATION OPTIMIZATION (CASE STUDY: PT. XYZ)***

**NOVIAN TAINDINI
NRP 5213 100 075**

**Supervisors
Wiwik Anggraeni S.Si, M.Kom**

**INFORMATION SYSTEMS DEPARTMENT
Information Technology Faculty
Sepuluh Nopember Institut of Technology
Surabaya 2017**

LEMBAR PENGESAHAN

PENERAPAN METODE KOMBINASI ALGORITMA GENETIKA DAN TABU SEARCH DALAM OPTIMASI ALOKASI KAPAL PETI KEMAS (STUDI KASUS : PT. XYZ)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

NOVIAN TIANDINI

NRP. 5213 100 075

Surabaya, 18 Januari 2017

**KETUA
JURUSAN SISTEM INFORMASI**

Dr. Ir. Aris Triyanto, M.Kom.

NIP.19650310 199102 1 001

LEMBAR PERSETUJUAN

PENERAPAN METODE KOMBINASI ALGORITMA GENETIKA DAN TABU SEARCH DALAM OPTIMASI ALOKASI KAPAL PETI KEMAS (STUDI KASUS : PT. XYZ)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

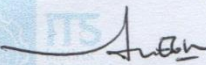
Oleh :

NOVIAN TIANDINI

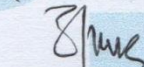
NRP. 5213 100 075

Disetujui Tim Penguji : Tanggal Ujian : 12 Januari 2017
Periode Wisuda : Maret 2017

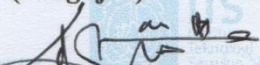
Wiwik Anggraeni, S.Si, M. Kom


(Pembimbing)

Edwin Riksakomara, S.Kom, M. T


(Penguji 1)

Faisal Mahananto, S.Kom, M.Eng, Ph.D


(Penguji 2)

PENERAPAN METODE KOMBINASI ALGORITMA GENETIKA DAN TABU SEARCH DALAM OPTIMASI ALOKASI KAPAL PETI KEMAS (STUDI KASUS : PT. XYZ)

Nama Mahasiswa : Novian Tiandini
NRP : 5213 100 075
Jurusan : Sistem Informasi FTIf - ITS
Dosen Pembimbing : Wiwik Anggraeni, S.Si, M.Kom

ABSTRAK

Perkembangan perdagangan global menyebabkan penggunaan jasa transportasi menjadi bagian yang sangat penting dalam pendistribusian barang. Salah satunya yaitu jasa transportasi laut atau jasa pelayaran. Saat ini banyak perusahaan – perusahaan produsen yang menggunakan jasa pelayaran untuk menyalurkan barang tersebut dari produsen sampai ke konsumen. Sehingga dengan terus berkembangnya jasa pelayaran, maka perlu adanya perencanaan dan keputusan-keputusan yang tepat dalam pengalokasian kapal yang akan digunakan dalam proses pengiriman barang. Oleh karena itu dalam tugas akhir ini akan dilakukan optimalisasi alokasi kapal dengan tujuan memaksimalkan profit dan memaksimalkan kapasitas. Pada penulisan tugas akhir ini di ambil studi kasus pada PT. XYZ dengan menggunakan metode kombinasi algoritma genetika dan tabu search.

Algoritma genetika adalah metode pemecahan masalah optimasi dengan menggunakan prinsip seleksi alam sedangkan tabu search adalah metode local search untuk memecahkan masalah local optima. Penggunaan metode kombinasi algoritma genetika dan tabu search pada pengalokasian kapal bertujuan untuk menemukan solusi yang optimum dalam mengalokasikan kapal. Berdasarkan perbandingan antara metode algoritma genetika (GA) dan

metode kombinasi algoritma genetika dan tabu search (GA-TS), diperoleh hasil profit dan muatan yang lebih optimal ketika menggunakan metode GA-TS dengan peningkatan profit sebesar 69% dan peningkatan load factor sebesar 14%. Peningkatan profit dan load factor juga ditunjukkan ketika dilakukan perbandingan antara kondisi pada perusahaan sebelum menerapkan GA-TS dan sesudah menerapkan GA-TS. Metode GA-TS memiliki profit dengan peningkatan lebih dari 100% dan peningkatan load factor sebesar 38% dibanding pada kondisi perusahaan. Sehingga berdasarkan hal ini, implementasi algoritma genetika dan tabu search dapat menjadi solusi bagi perusahaan dan membantu perusahaan dalam membuat perencanaan pengalokasian kapal. Kapal akan di alokasikan sesuai dengan muatan maksimal yang dapat di bawa dan profit maksimal yang akan diperoleh. Sehingga jumlah muatan peti kemas atau container yang dibawa oleh kapal menuju pelabuhan menjadi lebih optimal dan profit menjadi lebih maksimal.

Keywords : Optimasi, Alokasi, Jasa Pelayaran, Kapal Peti Kemas, Algoritma Genetika, Tabu Search

APPLICATION OF A HYBRID GENETIC ALGORITHM AND TABU SEARCH METHOD FOR CONTAINER SHIPS ALLOCATION OPTIMIZATION (CASE STUDY: PT. XYZ)

Name : Novian Tiandini
NRP : 5213 100 075
Department : Sistem Informasi FTIf - ITS
Supervisor : Wiwik Anggraeni, S.Si, M.Kom

ABSTRACT

The development of global trade led to the use of transport services became a very important part in the distribution of goods. One of them is sea transportation or shipping services. Currently, many companies - a manufacturer who uses shipping services to deliver the goods from producer to consumer. So with the continued development of shipping services, the need for planning and the right decisions in the allocation of vessels to be used in the delivery process. Therefore in this final task will be to optimize the allocation of vessels with the aim to maximize revenue and maximize capacity. In this thesis used a case study on PT. XYZ by using a combination of genetic algorithm and tabu search.

Genetic algorithm is a method of solving optimization problems using the principle of natural selection while the tabu search is a local search method to solve the problem of local optima. Use of a combination method of genetic algorithm and tabu search to the allocation of the ship, has a purpose to find the optimum solution in allocating vessel. Based on the comparison between genetic algorithms (GA) and a combination genetic algorithm and tabu search (GA-TS), the result of profit and load factor are optimized when using GA-TS with profit increases 69% and an increase in load factor of 14%. Increased profit and load factor is also

indicated when the comparison between the conditions in the company before applying the GA-TS and after applying the GA-TS. GA-TS methods has a profit with an increase of over 100% and an increase in load factor of 38% compared to the condition of the company. So based on this, the implementation of genetic algorithm and tabu search can be a solution for the company and help companies in planning the allocation of the ship. The vessel will be allocated in accordance with the maximum charge that can be brought and the maximum profit to be obtained. So that the number of cargo or container carried by ships heading for the port to become more optimal and profit be maximal.

Keywords: Optimization, Allocation, Shipping Services, Ship Container, Genetic Algorithms, Tabu Search

KATA PENGANTAR

Alhamdulillah atas karunia, rahmat, barakah, dan jalan yang telah diberikan Allah SWT selama ini sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir dengan judul:

PENERAPAN METODE KOMBINASI ALGORITMA GENETIKA DAN TABU SEARCH DALAM OPTIMASI ALOKASI KAPAL PETI KEMAS (STUDI KASUS : PT. XYZ)

Terima kasih atas pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik materi maupun spiritual demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Allah SWT yang telah memberikan kesempatan, kemudahan dan petunjuk kepada penulis dalam menyelesaikan Tugas Akhir.
2. M. Hosnan Amin dan R.Noerhayati selaku orang tua penulis telah mendokan dan mendukung sehingga penulis dapat menyelesaikan buku tugas akhir dengan maksimal.
3. Ibu Wiwik Anggraeni, S.Si, M. Kom. selaku dosen pembimbing yang meluangkan waktu, memberikan ilmu, petunjuk, dan motivasi untuk kelancaran menyusun Tugas Akhir ini.
4. Bapak Edwin Riksakomara, S.Kom, M. T dan Bapak Faisal Mahananto, S.Kom, M.Eng, Ph.D selaku dosen penguji yang telah memberikan masukan untuk penyempurnaan pengerjaan tugas akhir.
5. Dr. Apol Pribadi Subriadi, S.T., M.T selaku dosen wali, terima kasih atas motivasi selama penulis menjadi mahasiswa di Jurusan Sistem Informasi.
6. Seluruh dosen Jurusan Sistem Informasi ITS yang telah membagikan seluruh ilmu yang sangat berharga.

7. Mas Ricky Asrul Sani selaku admin dan Asvin Immadudin selaku aslab yang telah memberikan tempat dan kenyamanan di laboratorium Rekayasa Data dan Itelegensi Bisnis.
8. Surya Hadi Wijaya selaku teman yang telah banyak membantu, mengajari dan memberikan arahan dalam mengerjakan tugas akhir.
9. Fajar Ratna, Mia Eka, Sofi Arianti, Elisa Dian, Rifatun, Provani, Aninidita, Kamal, Slamet dan Maulana selaku teman seperjuangan yang saling memberi semangat dan motivasi.
10. Teman – teman BELTRANIS yang selalu menemani penulis dalam melewati hari hari bersama dan berbagi pengalaman
11. Seluruh mahasiswa jurusan Sistem Informasi yang membantu dalam membentuk pribadi yang tangguh dan bermanfaat
12. Berbagai pihak yang membantu dalam penyusunan Tugas Akhir ini dan belum dapat disebutkan satu per satu.

Penyusunan laporan ini masih jauh dari sempurna, untuk itu saya menerima adanya kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
LEMBAR PERSETUJUAN	iv
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah	3
1.3. Batasan Pengerjaan Tugas Akhir	4
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Relevansi	5
1.7. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA	9
2.1 Penelitian Sebelumnya	9
2.2 Dasar Teori	12
2.2.1 Optimasi.....	12
2.2.2 Alokasi Sumberdaya.....	13
2.2.3 Jasa Pelayaran.....	13
2.2.4 Algoritma Genetika	14
2.2.5 Tabu Search.....	19
2.2.6 <i>Standard Deviasi</i>	22
2.2.7 <i>Error Variance</i>	23
BAB III METODOLOGI PENELITIAN.....	25
3.1 Urutan pelaksanaan proses penelitian	25
3.1.1 Identifikasi Masalah.....	26
3.1.2 Perumusan Masalah	26
3.1.3 Studi Pustaka.....	26
3.1.4 Pengumpulan Data.....	26
3.1.5 Pemodelan Masalah dalam Bentuk Matematis....	27

3.1.6	Perancangan Model Algoritma Genetika dan <i>Tabu Search</i>	27
3.1.7	Implementasi Algoritma Genetika dan <i>Tabu Search</i>	27
3.1.8	Evaluasi dan Revisi	28
3.1.9	Analisa Hasil dan Penarikan Kesimpulan	28
3.1.10	Penyusunan Laporan Tugas Akhir	28
3.2	Bahan dan peralatan yang digunakan.....	28
3.2.1	Bahan yang Digunakan.....	28
3.2.2	Alat yang Digunakan	29
BAB IV PERANCANGAN		31
4.1	Pemodelan Permasalahan dalam Bentuk Matematis....	31
4.1.1	Deskripsi Data.....	31
4.1.2	Penentuan Variabel.....	33
4.1.3	Perumusan Fungsi <i>Fitness</i>	34
4.1.4	Perumusan Batasan Model.....	34
4.2	Perancangan Algoritma Genetika dan <i>Tabu Search</i>	36
4.2.1	Inisialisasi Parameter	37
4.2.2	Bangkitkan Popoulasi	39
4.2.3	Evaluasi Populasi	41
4.2.4	Elitisme	42
4.2.5	Seleksi Kromosom.....	42
4.2.6	<i>Crossover</i>	42
4.2.7	Mutasi.....	43
4.2.8	Penerapan <i>Tabu Serach</i>	43
BAB V IMPLEMNTASI		45
5.1	Inisialisasi parameter.....	45
5.2	Bangkitkan Populasi	47
5.3	Evaluasi Populasi.....	50
5.4	Elitisme	52
5.5	Seleksi Kromosom.....	52
5.6	<i>Crossover</i>	53
5.7	Mutasi.....	54
5.8	<i>Tabu Search</i>	56
5.8.1	Inisialisasi Parameter	56
5.8.2	Inisialisasi Solusi	57
5.8.3	Menghasilkan Solusi Terbaik.....	57

5.8.4	Update Solusi dan Tabu List	58
BAB VI HASIL DAN PEMBAHASAN.....		61
6.1	Hasil Percobaan	61
6.1.1	Lingkungan Uji Coba.....	61
6.1.2	Parameter dan Sekenario Uji Coba	62
6.1.3	Hasil Uji Coba	63
6.1.4	Sekenario Uji Coba Terbaik.....	67
6.1.5	Proses <i>Running</i> GA dan GA-TS.....	67
6.1.6	Validasi	70
6.2	Pembahasan dan Analisis	73
6.2.1	Analisis Hasil Uji Coba Algoritma Genetika	73
6.2.2	Analisis Hasil Uji Coba Kombinasi Algoritma Genetika dan <i>Tabu Search</i>	77
6.2.3	Perbandingan Hasil Algoritma Genetika dengan Algoritma Genetika dan <i>Pencarian Tabu</i>	80
6.2.4	Perbandingan sebelum dan sesudah implementasi GA-TS	82
BAB VII KESIMPULAN DAN SARAN		85
7.1	Kesimpulan.....	85
7.2	Saran.....	86
DAFTAR PUSTAKA		89
BIODATA PENULIS		93
LAMPIRAN A		A-1
LAMPIRAN B		B-1
LAMPIRAN C		C-1
LAMPIRAN D		D-1
LAMPIRAN E		E-1
LAMPIRAN F		F-1

DAFTAR GAMBAR

Gambar 2.1 Diagram Alir Algoritma Genetika	15
Gambar 3.1 Metode Penelitian	25
Gambar 4.1 <i>Flowchart GA-TS</i>	37
Gambar 4.2 Gen Bagian dari Kromosom.....	40
Gambar 4.3 Potongan Kromosom	41
Gambar 4.4 Populasi.....	41
Gambar 5.1 Inisialisasi Parameter	46
Gambar 5.2 Pembentukan Header	47
Gambar 5.3 Hasil Header.....	47
Gambar 5.4 Bangkitkan Populasi	49
Gambar 5.5 Update Populasi.....	50
Gambar 5.6 Evaluasi Populasi.....	51
Gambar 5.7 Profit Aktual.....	51
Gambar 5.8 Kromosom Terpilih	52
Gambar 5.9 Elitisme	52
Gambar 5.10 Proses Lottery.....	53
Gambar 5.11 Proses Seleksi	53
Gambar 5.12 Proses Crossover.....	54
Gambar 5.13 Proses Mutasi	55
Gambar 5.14 Pemilihan Gen	56
Gambar 5.15 Inisialisasi Parameter	56
Gambar 5.16 Inisialisasi Solusi	57
Gambar 5.17 Solusi Terbaik.....	58
Gambar 5.18 Update Tabu list dan Solusi.....	59
Gambar 8.1 Data Kapasitas Kapal.....	C-1
Gambar 8.2 Data Harga Sewa Kontainer	C-2
Gambar 8.3 Data Permintaan	C-3
Gambar 8.4 Hasil Gen Maksimal dan Minimal.....	D-1
Gambar 8.5 Hasil GA Gen 1-60	E-1
Gambar 8.6 Hasil GA Gen 61-135	E-2
Gambar 8.7 Hasil GA Gen 136 - 195	E-3
Gambar 8.8 Hasil Profit TS.....	F-1
Gambar 8.9 Hasil Muatan TS.....	F-2

DAFTAR TABEL

Tabel 2.1 Penelitian Sebelumnya.....	9
Tabel 4.1 Deskripsi Data.....	32
Tabel 6.1 Lingkunga Uji Coba	61
Tabel 6.2 Uji Coba GA 1	63
Tabel 6.3 Uji Coba GA 2	64
Tabel 6.4 Uji Coba GA 3	64
Tabel 6.5 Uji Coba GA 4	65
Tabel 6.6 Uji Coba GA-TS 1.....	65
Tabel 6.7 Uji Coba GA-TS 2.....	65
Tabel 6.8 Uji Coba GA-TS 3.....	66
Tabel 6.9 Uji Coba GA-TS 4.....	66
Tabel 6.10 Sekenario Terbaik.....	67
Tabel 6.11 Running GA	68
Tabel 6.12 Running GA-TS	69
Tabel 6.13 Profit Data Perusahaan.....	71
Tabel 6.14 Validasi GA-TS.....	72
Tabel 6.15 GA Muatan Tujuan 1	74
Tabel 6.16 GA Muatan Tujuan 2	75
Tabel 6.17 GA Muatan Tujuan 3	75
Tabel 6.18 GA Muatan Tujuan 4.....	76
Tabel 6.19 GA Profit.....	76
Tabel 6.20 Load Factor hasil GA.....	77
Tabel 6.21 GA-TS Tujuan 1	78
Tabel 6.22 GAT-TS Tujuan 2.....	78
Tabel 6.23 GA-TS Tujuan 3	79
Tabel 6.24 GA-TS Tujuan 4.....	79
Tabel 6.25 GA-TS Profit.....	80
Tabel 6.26 Load Factor hasil GA-TS	80
Tabel 6.27 Perbandingan Profit GA & GA-TS.....	81
Tabel 6.28 Perbandingan Load Factor GA dan GA-TS	82
Tabel 6.29 Perbandingan Sbelum dan Sesudah Optimasi	83
Tabel 8.1 Lokasi 1 – Makassar GA	A-1
Tabel 8.2 Lokasi 2 – Bitung GA.....	A-6
Tabel 8.3 Lokasi 3 –Ambon GA.....	A-8

Tabel 8.4 Lokasi 4 – Banjarmasin GA.....	A-10
Tabel 8.5 Lokasi 1 – Makassar GA-TS.....	B-1
Tabel 8.6 Lokasi 2 – Bitung GA-TS	B-4
Tabel 8.7 Lokasi 3 –Ambon GA-TS.....	B-5
Tabel 8.8 Lokasi 4 – Banjarmasin GA-TS	B-6

BAB I

PENDAHULUAN

Bab ini menjelaskan tentang pendahuluan pengerjaan tugas akhir, yang meliputi latar belakang, rumusan permasalahan, batasan masalah, tujuan penelitian hingga manfaat yang diperoleh dari penelitian ini.

1.1. Latar Belakang

Indonesia merupakan negara kepulauan terbesar di dunia yang terdiri dari 17.508 pulau dan memiliki luas wilayah laut yang sangat luas yaitu 7,900,000 km² [1]. Letak geografis Indonesia yang strategis serta besarnya potensi ekonomi kelautan dapat memberi kesempatan yang besar bagi Indonesia untuk mempercepat terwujudnya kemandirian ekonomi nasional. Pelayaran atau angkutan laut adalah salah satu jasa yang memiliki peranan penting dalam mendorong dan membantu menghadapi perubahan dan perkembangan ekonomi nasional yaitu melalui perdagangan global. Sejalan dengan perkembangan perdagangan global, transportasi laut menjadi salah satu bagian yang sangat penting dalam pendistribusian barang. Saat ini banyak perusahaan – perusahaan produsen yang menggunakan jasa pelayaran untuk menyalurkan barang tersebut dari produsen sampai ke konsumen atau pemakai industri.

Selain itu perusahaan penyedia jasa pelayaran juga memiliki tanggung jawab dalam pengalokasian kapal yang akan digunakan dalam proses pengiriman barang. Sehingga dengan terus berkembangnya jasa pelayaran dan tanggung jawab yang dimiliki, maka perlu adanya perencanaan dan keputusan-keputusan yang tepat dalam menyediakan jasa pelayaran agar perusahaan penyedia jasa dapat terus bersaing dan memeberikan pelayanan yang terbaik. Oleh karena itu dalam tugas akhir ini akan dilakukan optimalisasi alokasi kapal dengan tujuan memaksimalkan keuntungan dan

memaksimalkan kapasitas muat kapal. Pada penulisan tugas akhir ini di ambil studi kasus pada PT. XYZ.

PT. XYZ adalah perusahaan pelayaran yang menyediakan one stop service dalam memberikan pelayanan logistik kepada para pelanggan di seluruh kepulauan Indonesia. Dalam menjalankan roda usahanya dalam bidang transportasi khususnya jasa transportasi laut, PT. XYZ menyediakan pelayanan penyediaan pengiriman barang dalam peti kemas melalui jalur laut. Transportasi laut tersebut berada di pelabuhan-pelabuhan utama dengan pusat pelabuhan berada di Jakarta dan Surabaya. Adapun komoditas utama yang dikemas, antara lain: beras, tepung, gula, semen dan muatan-muatan umum yang menggunakan kontainer. Komoditas-komoditas ini dahulunya menggunakan kapal muatan umum dan dilakukan secara masal, namun atas permintaan keamanan, efisiensi dan efektivitas, pengangkutannya kini lebih banyak melalui peti kemas [2].

Sejauh ini PT. XYZ masih menggunakan cara manual dalam penyusunan perencanaan pengalokasian kapal. Sehingga perlu adanya tugas akhir dengan menggunakan kombinasi algoritma genetika dan tabu search sebagai sebuah metode yang dapat membantu dalam menentukan perencanaan pengalokasian kapal terbaik berdasarkan tujuan yang ingin di capai. Diharapkan dengan adanya perencanaan pengalokasian kapal dapat membantu proses pengiriman barang menjadi lancar dengan *profit* yang dimaksimalkan.

Metode yang akan digunakan pada tugas akhir ini yaitu metode kombinasi algoritma genetika dan tabu search. Algoritma genetika yang dikembangkan oleh Goldberg adalah algoritma komputasi yang diinspirasi teori evolusi Darwin yang menyatakan bahwa kelangsungan hidup suatu makhluk dipengaruhi aturan “yang kuat adalah yang menang” [3]. Kelebihan metode algoritma genetika dibandingkan dengan metode optimasi lainnya adalah algoritma genetika merupakan algoritma berbasis populasi yang memungkinkan digunakan

pada optimasi masalah dengan ruang pencarian (search space) yang sangat luas dan kompleks dan metode ini bisa diimplementasikan pada berbagai macam data seperti data yang dibangkitkan secara numerik atau menggunakan fungsi analitis [4]. Namun algoritma genetika juga memiliki kekurangan, yaitu dibutuhkan iterasi yang lebih lama dan algoritma genetika cenderung terjebak ke dalam solusi local optima [5], [6]. Sehingga untuk menutupi kekurangan dalam mengidentifikasi solusi local optima digunakan metode tabu search untuk menghindari lingkaran di bidang solusi yang sama, selain itu tabu search memiliki kecepatan eksekusi lebih tinggi dari pada algoritma genetika karena tabu search tidak kembali pada solusi yang sudah dieksplorasi [6].

Penggunaan metode kombinasi algoritma genetika dan tabu search dapat menghasilkan nilai fitness yang lebih baik dengan jumlah iterasi yang lebih sedikit dibandingkan menggunakan metode algoritma genetika saja [7]. Sehingga pada penelitian tugas akhir ini dilakukan kombinasi metode algoritma genetika dan tabu search yang dapat saling mendukung untuk menemukan solusi optimum. Terdapat beberapa keterkaitan penelitian terdahulu dengan penelitian tugas akhir ini di antaranya yaitu penelitian A novel hybrid genetic algorithm with tabu search for optimizing multi-dimensional functions and point pattern recognition [6], An improved Genetic Algorithm for spatial optimization of multi-objective and multi-site land use allocation [8], Kombinasi algoritma genetika dan tabu search dalam Pembuatan Tabel Jadwal Mata Kuliah [7]. Maka dari itu metode kombinasi algoritma genetika dan tabu search sangat cocok untuk digunakan dalam studi kasus pada penelitian tugas akhir ini.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, maka dapat disusun rumusan masalah sebagai berikut :

1. Bagaimana menentukan alokasi kapal yang optimal dalam proses pengiriman barang menuju pelabuhan –

pelabuhan di Indonesia dengan tujuan memaksimalkan *profit* dan memaksimalkan kapasitas kapal menggunakan metode kombinasi algoritma genetika dan *tabu search* ?

2. Bagaimana perbandingan anantara sebelum dan sesudah melakukan optimasi ?
3. Bagaimana hasil rekomendasi rencana pengalokasian kapal berdasarkan metode kombinasi algoritma genetika dan *tabu search*?

1.3. Batasan Pengerjaan Tugas Akhir

Batasan masalah pada tugas akhir ini adalah :

1. Optimasi yang dilakukan berfokus pada pengalokasian kapal peti kemas pada PT. XYZ dengan tujuan memaksimalkan *profit* dan memaksimalkan kapasitas kapal.
2. Pengalokasian kapal dilakukan terbatas pada kapal peti kemas dalam negeri miliki PT. XYZ

1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah sebagai berikut :

1. Menentukan dan mengalokasikan kapal yang optimum dalam proses pengiriman barang menuju pelabuhan – pelabuhan di Indonesia dengan tujuan memaksimalkan *profit* dan memaksimalkan kapasitas kapal menggunakan metode kombinasi algoritma genetika dan *tabu search*
2. Mengetahui perbandingan anantara sebelum dan sesudah melakukan optimasi.
3. Memberikan hasil rekomendasi rencana pengalokasian kapal berdasarkan metode kombinasi algoritma genetika dan *tabu search*.

1.5 Manfaat Penelitian

Diharapkan dengan adanya penelitian ini akan diperoleh manfaat bagi perusahaan secara umum terkait dengan proses produksi sebagai berikut;

1. Bagi perusahaan :

Manfaat dari tugas akhir ini diharapkan dapat memudahkan PT. XYZ untuk menentukan dan mengalokasikan kapal peti kemas yang optimum dalam proses pengiriman barang dengan tujuan memaksimalkan *profit* dan memaksimalkan kapasitas kapal.

2. Bagi pengembangan keilmuan :

Manfaat dari tugas akhir ini diharapkan dapat dijadikan sebagai pengetahuan dalam melakukan optimalisasi alokai sumber daya dan dapat dijadikan pustaka untuk penelitian berikutnya.

1.6 Relevansi

Pengguna jasa pelayaran bukan lagi hal yang minoritas di Indonesia, sudah banyak perusahaan yang memanfaatkan jasa pelayaran sebagai jasa transportasi pengiriman barang untuk memenuhi proses bisnis perusahaan. Banyaknya pengguna jasa pelayaran menyebabkan perusahaan pelayaran terus bersaing dalam menyediakan jasa terbaik sehingga penyedia pelayaran terus melakukan pengembangan. Untuk dapat terus bersaing maka perlu adanya perencanaan dan keputusan – keputusan yang tepat dalam menyediakan jasa pelayaran. Salah satunya yaitu dalam penentuan kapal yang akan digunakan dalam proses pengiriman barang. Untuk dapat membuat perencanaan pengalokasian kapal yang tepat dan optimum, maka perlu dilakukan optimasi alokasi kapal. Penerapan metode kombinasi algoritma genetika dan *tabu search* dapat membantu dalam pengalokasian kapal yang optimum. Sehingga hasil dari penelitian ini dapat bermanfaat bagi pengguna maupun penyedia jasa pelayaran yang nantinya dapat berpengaruh pada perkembangan ekonomi Indonesia.

1.7 Sistematika Penulisan

Sistematika penulisan laporan disesuaikan dengan pelaksanaan pengerjaan tugas akhir yang dibagi menjadi 7 bab sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisikan hal-hal yang mendorong atau melatarbelakangi pentingnya dilakukan tugas akhir ini dengan komponen-komponen, yaitu latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, relevansi penelitian terhadap bidang keilmuan, serta sistematika penulisan yang diterapkan dalam memaparkan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini terdiri dari penjelasan mengenai studi sebelumnya yang mendukung tugas akhir dan konsep atau teori-teori yang memiliki keterkaitan terhadap topik yang diangkat, yaitu proses bisnis perusahaan, konsep optimasi, konsep algoritma genetika dan konsep *tabu search*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi penjelasan mengenai masing-masing tahapan dalam pembuatan tugas akhir, mulai dari identifikasi masalah, perumusan masalah, studi pustaka, pemodelan masalah dalam bentuk matematis, perancangan model algoritma genetika dan *tabu search*, implementasi algoritma genetika dan *tabu search*, evaluasi dan revisi, analisa hasil dan penarikan kesimpulan dan yang terakhir yaitu penyusunan tugas akhir.

BAB IV PERANCANGAN

Pada bab ini membahas terkait perancangan pemodelan masalah alokasi kapal peti kemas dalam bentuk matematis yang di modelkan dengan proses algoritma genetika dan *tabu search* yang dimulai dari pengumpulan data hingga mendapatkan model optimasi alokasi kapal peti kemas yang paling optimal.

BAB V IMPLEMENTASI

Bab ini menjelaskan mengenai proses pelaksanaan penelitian dari mulai implementasi model algoritma genetika yang sudah di rancang pada tahap sebelumnya dan implementasi program komputer berdasarkan algoritma yang ada. Impelementasi ini meliputi implementasi data dan proses berdasarkan perancangan pada tahap sebelumnya.

BAB VI HASIL DAN PEMBAHASAN

Pada bab ini dilakukan analisis terhadap hasil yang didapat dari uji coba yang telah dilakukan pada tahap sebelumnya yaitu implementasi algoritma genetika dan *tabu search* yang diharapkan dapat menjadi penyelesaian permasalahan yang dibahas pada tugas akhir ini.

BAB VII KESIMPULAN DAN SARAN

Bab terakhir sebagai penutup dari laporan tugas akhir yang berisi kesimpulan dan saran dari seluruh percobaan yang telah dilakukan untuk dibandingkan dengan tujuan dan permasalahan yang telah dijabarkan pada bab Pendahuluan.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Bab ini dijelaskan mengenai penelitian sebelumnya dan dasar teori yang akan digunakan sebagai acuan atau landasan dalam pengerjaan tugas akhir ini. Landasan teori akan memberikan gambaran secara umum dari landasan penjabaran tugas akhir. Penelitian terdahulu dan landasan teori yang digunakan merupakan penelitian dan teori yang berhubungan dengan pengerjaan tugas akhir.

2.1 Penelitian Sebelumnya

Dalam proses pengerjaan tugas akhir ini, dilakukan pencarian beberapa penelitian sebelumnya untuk dijadikan referensi dalam pengerjaan. Sehingga dengan adanya referensi tersebut dapat memperbaiki kesalahan kesalahan dalam pengerjaan yang ada dan dapat membantu pengerjaan tugas akhir dengan lebih efisien. Pada tugas akhir ini dilakukan penelitian mengenai optimalisasi alokasi sumber daya berupa kapal yang akan di alokasikan dalam proses pengiriman barang menggunakan metode kombinasi algoritma genetika dan *tabu seacrh*. Berikut pada Tabel 2.1 merupakan beberapa penelitian yang dijadikan referensi.

Tabel 2.1 Penelitian Sebelumnya

Penelitian 1 [1]	
Judul Penelitian	A novel hybrid genetic algorithm with Tabu search for optimizing multi-dimensional functions and point pattern recognition
Penulis/ Tahun Penelitian	Gautam Garai, B.B. Chaudhurii / 2013

Penelitian 1 [1]	
Gambaran Umum	Pada penelitian ini menyajikan pendekatan evolusioner hybrid baru untuk mengoptimalkan fungsi matematika dan masalah Point Pattern Recognition (PPR). Metode yang diusulkan menggabungkan algoritma genetika dan pencarian tabu. Pendekatan ini dapat mencapai global optimum atau near-global dalam ruang dimensi yang tinggi.
Keterkaitan Tugas Akhir	Keterkaitan dengan tugas akhir adalah penggunaan metode yang sama yaitu kombinasi algoritma genetika dan <i>tabu search</i> sehingga dapat menjadi acuan dalam penerapan metode tersebut.
Penelitian 2 [2]	
Judul Penelitian	Combined ship allocation, routing and freight assignment in tramp shipping
Penulis/ Tahun Penelitian	Dung-Ying Lin , Hui-Yen Liu / 2011
Gambaran Umum	Pada penelitian ini metode yang digunakan yaitu metode algoritma genetika untuk mengalokasikan kapal tramp dengan mempertimbangkan rute dan penugasan kapal

Penelitian 2 [2]	
Keterkaitan Tugas Akhir	Keterkaitan dengan tugas akhir adalah adanya topik yang sama yaitu mengenai pengalokasian kapal menggunakan metode algoritma genetika. Namun pada penelitian ini mempertimbangkan rute dan penugasan kapal.
Penelitian 3 [3]	
Judul Penelitian	Implementasi algoritma genetika dalam Pengembangan Sistem Aplikasi Penjadwalan Kuliah
Penulis/ Tahun Penelitian	Komang Setemen / 2010
Gambaran Umum	Pada penelitian ini dilakukan pemecahan masalah penjadwalan kuliah di perguruan tinggi dengan menggunakan metode meta-heuristic, yaitu: algoritma genetika dan <i>tabu search</i> .
Keterkaitan Tugas Akhir	Keterkaitan dengan tugas akhir adalah penggunaan metode yang sama yaitu kombinasi metode algoritma genetika dan <i>tabu search</i> yang digunakan untuk melakukan penjadwalan yang dapat juga disebut penempatan atau pengalokasian sumber daya. Sehingga penelitian ini dapat dijadikan sebagai

	tinjauan pustaka dalam penyusunan tugas akhir.
Penelitian 4 [4]	
Judul Penelitian	Optimasi Penjadwalan Mata Kuliah Dengan Algoritma Genetika (Studi Kasus di AMIK JTC Semarang)
Penulis/ Tahun Penelitian	Entot Suhartono / 2015
Gambaran Umum	Pada penelitian ini metode yang digunakan adalah metode algoritma genetika untuk mengalokasikan ruang kuliah dan tenaga kerja dosen
Keterkaitan Tugas Akhir	Keterkaitan dengan tugas akhir adalah penggunaan metode yang sama dan topik yang sama yaitu penggunaan metode algoritma genetika untuk mengalokasikan sumber daya.

2.2 Dasar Teori

Pada bagian ini menjelaskan mengenai dasar teori dari penulisan tugas akhir ini, yang berfungsi agar pembaca dapat memahami maksud dari penelitian ini dengan mudah. Dasar teori yang digunakan dapat berasal dari jurnal ilmiah, buku ataupun artikel ilmiah.

2.2.1 Optimasi

Persoalan optimasi (*optimization problem*) adalah persoalan yang menuntut pencarian solusi optimum [5]. Persoalan optimasi dibagi menjadi dua macam, yaitu maksimasi (*maximization*) dan minimasi (*minimization*). Ada dua metode dalam penyelesaian masalah optimasi, yaitu (1) Metode

Konvensional yang diterapkan dengan menggunakan perhitungan matematika murni atau secara biasa. Ada beberapa metode konvensional yang sering digunakan untuk menyelesaikan masalah optimasi, diantaranya: algoritma Dijkstra, algoritma Floyd-Warshall, dan algoritma Bellman-Ford. (2) Metode Heuristik salah satu dari bidang kecerdasan buatan yang digunakan untuk menyelesaikan masalah optimasi. Terdapat beberapa algoritma dari metode heuristik yang sering digunakan dalam permasalahan optimasi, diantaranya adalah algoritma genetika, algoritma pencarian tabu, jaringan saraf tiruan, algoritma semut dan lain-lain [4]

2.2.2 Alokasi Sumberdaya

Resource allocation problem (RAP) adalah proses mengalokasikan sumber daya di antara berbagai proyek atau unit bisnis untuk memaksimalkan keuntungan atau meminimalkan biaya. Proses RAP berusaha untuk menemukan alokasi optimal dari jumlah sumber daya yang terbatas dengan sejumlah tugas untuk mengoptimalkan tujuan mereka dengan batasan atau kendala yang diberikan. Sumber daya mungkin berupa orang, aset, bahan, atau modal yang dapat digunakan untuk mencapai tujuan [6].

2.2.3 Jasa Pelayaran

Usaha pelayaran adalah kegiatan yang menawarkan jasa pengangkutan melalui laut dalam bentuk pengangkutan barang dari pelabuhan pemuatan sampai ke pelabuhan bongkar atau tujuan dalam keadaan baik atau sesuai dengan perjanjian antara pengangkut dan penjual atau penerima barang [7].

Secara umum ada tiga tipe pelayaran di transportasi maritim yaitu pelayaran liner, pelayaran *tramper* atau *charter*, dan pelayaran industrial [8]. Pelayaran liner beroperasi seperti bus umum, kapal biasanya beroperasi pada rute yang tetap dan setiap kapal memiliki jadwal masing-masing pada pelabuhan di rute yang dilaluinya. Muatan pada kapal hampir selalu penuh karena melakukan kegiatan bongkar muat pada tiap pelabuhan yang dilewati. Pelayaran *tramper* beroperasi seperti

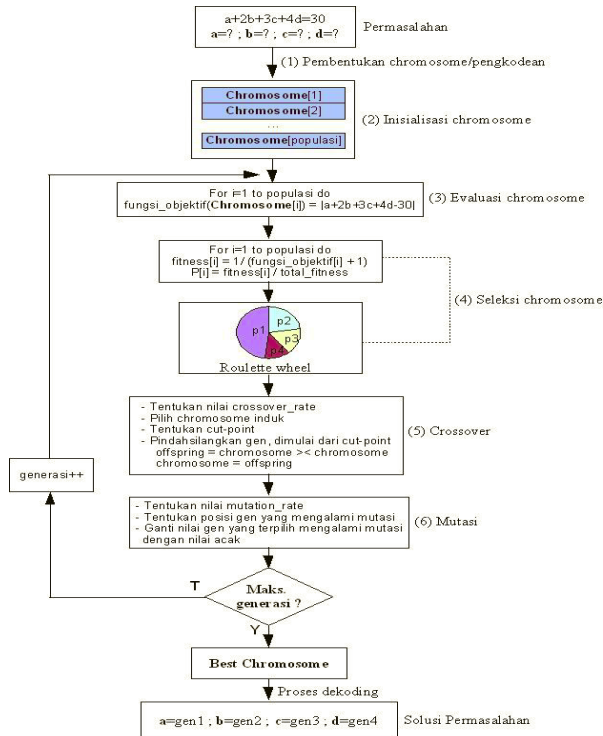
sebuah taxi, kapal baru akan beroperasi ketika ada permintaan dari pemilik barang. Lalu kapal berangkat saat muatan tersedia dan biasanya kapasitas muatan kapal terisi penuh hanya untuk satu tujuan. Sedangkan pelayaran industrial beroperasi seperti armada truk di daratan. Pemilik muatan biasanya adalah perusahaan besar yang mengangkut muatannya sendiri. Pemilik muatan dapat mengatur seluruh kegiatan armada kapal. Kapal biasanya dimiliki secara langsung atau menyewa [9]

2.2.4 Algoritma Genetika

Algoritma genetika merupakan salah satu metode untuk menentukan optimalisasi atas dasar Teori Darwin. Langkah prosedur algoritma ini diawali dengan menentukan suatu set solusi potensial dan melakukan perubahan dengan beberapa perulangan (iterasi) dengan algoritma genetika untuk menghasilkan solusi terbaik. Set solusi potensial ini ditetapkan diawal dan disebut dengan kromosom. Kromosom ini dibentuk secara random berupa susunan angka binary yang di-generate dan dipilih. Keseluruhan set dari kromosom yang diobservasi mewakili suatu populasi [10].

Kromosom-kromosom berevolusi beberapa kali tahapan iterasi yang disebut dengan generasi. Generasi baru (*offsprings*) degenerate dengan teknik kawin silang (*crossover*) dan mutasi (*mutation*). *Crossover* meliputi pemisahan atau pemecahan (*splitting*) dua kromosom, kemudian mengkombinasikan setengah bagian dari masing-masing kromosom dengan pasangan-pasangan lainnya. Mutasi meliputi pertukaran (*flipping*) satu bit (bagian) dari kromosom dengan satu bagian lain dari kromosom lain yang menjadi pasangannya. Kromosom-kromosom tersebut kemudian berevolusi dengan suatu kriteria kesesuaian (*fitness*) yang telah ditetapkan, hasil yang terbaik akan dipilih sedangkan yang lainnya diabaikan. Proses ini dilakukan secara berulangulang sampai menemukan suatu kromosom yang memiliki kesesuaian terbaik (*best fitness*) untuk dijadikan sebagai solusi terbaik dari suatu

masalah [4]. Gambar 2.1 adalah diagram penerapan algoritma genetika yang digunakan untuk menyelesaikan masalah kombinasi [11]



Gambar 2.1 Diagram Alir Algoritma Genetika

Terdapat enam komponen utama yang terdapat di dalam algoritma genetika [12], yaitu :

A. Teknik Pengkodean

Suatu teknik bagaimana mengkodekan gen dari kromosom. Teknik ini merupakan teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok masalah. Teknik pengkodean meliputi pengkodean gen dan kromosom. Gen adalah bagian dari kromosom

yang dapat direpresentasikan dalam bentuk string bit, tree, array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lain yang dapat diimplementasikan untuk operator genetika.

B. Prosedur Inisialisasi (generate populasi awal)

Suatu proses yang menghasilkan sejumlah individu secara acak (random). Banyaknya populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diterapkan. Setelah jumlah populasi ditentukan, selanjutnya dilakukan inisialisasi terhadap kromosom yang ada di dalam populasi tersebut. Inisialisasi kromosom dilakukan secara acak, dengan tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

C. Fungsi Evaluasi

Individu dievaluasi berdasarkan fungsi tertentu sebagai ukuran kinerjanya. Individu dengan nilai *fitness* tinggi pada kromosomnya yang akan dipertahankan, sedangkan individu yang pada kromosomnya bernilai *fitness* rendah akan diganti. Fungsi *fitness* tergantung pada permasalahan tertentu dari representasi yang digunakan [13]. Fungsi *fitness* bisa berhubungan langsung dengan fungsi tujuan, atau bisa juga sedikit modifikasi terhadap fungsi tujuan. Sejumlah solusi yang dibangkitkan dalam populasi akan dievaluasi menggunakan fungsi *fitness*. Fungsi *fitness* yang biasa digunakan adalah $F(x) = \frac{1}{1+f(x)}$, dimana $f(x)$ adalah fungsi *fitness* dari permasalahan yang ingin diselesaikan. Untuk kasus minimasi, jika didapatkan $f(x)$ yang kecil maka nilai *fitness* akan besar. Sebaliknya, untuk kasus maksimasi fungsi *fitness* bisa menggunakan nilai $F(x) = f(x)$ [14].

D. Seleksi

Proses seleksi bertujuan untuk memilih individu-individu yang akan dipilih untuk proses persilangan dan mutasi, sehingga akan diperoleh calon induk yang baik. Induk yang baik akan menghasilkan keturunan yang baik. Langkah pertama dalam seleksi yaitu pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya. Ada beberapa metode untuk memilih kromosom yang digunakan antara lain :

a. *Roulette wheel selection*

Seleksi ini dilakukan dengan cara menyeleksi parent dengan tujuan untuk mempertahankan nilai *fitness*-nya agar memiliki kesempatan untuk diseleksi adalah kromosom yang baik. Proses ini diibaratkan seperti permainan roda rolet (*roulette wheel*), di mana semua kromosom ditempatkan dalam populasi, setiap tempat besar sesuai dengan fungsi *fitness*. Kromosom dipilih berdasarkan nilai *fitness*, semakin besar nilai *fitness* maka kromosom tersebut mempunyai peluang untuk dipilih beberapa kali. Metode seleksi roda rolet merupakan metode yang paling sederhana, dan sering juga dikenal dengan nama *stochastic sampling with replacement*.

b. Seleksi *good fitness*

Seleksi ini dilakukan dengan cara setengah dari jumlah populasi yang memiliki nilai *fitness* yang paling rendah akan dihilangkan, sehingga selalu hanya tersisa sekelompok solusi yang terbaik. Solusi yang tersisa hasil dari seleksi tersebut disebut populasi induk. Karena jumlah populasi harus tetap, maka perlu di-generate solusi baru sebanyak setengah dari jumlah populasi yang

ada. Ada 2 cara yang digunakan untuk men-generate solusi baru, yaitu dengan cara reproduksi kromosom baru dan mutasi dari solusi induk. Tujuan men-generate solusi baru ini adalah untuk menemukan alternatif solusi yang lebih baik dari solusi-solusi yang sudah dihasilkan.

E. Operator Genetika

Algoritma genetika merupakan proses pencarian yang heuristik dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan keberhasilan algoritma genetik dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harus diperhatikan adalah menghindari terjadinya konvergensi premature, yaitu mencapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal. Ada dua operator genetika yaitu:

a. *Crossover* (Persilangan)

Crossover merupakan proses di dalam algoritma genetika yang bekerja untuk meng-gabungkan dua kromosom parent menjadi kromosom baru (offspring) pada suatu waktu. Sebuah kromosom yang mengarah pada solusi baik dapat diperoleh melalui proses *crossover* pada dua buah kromosom. Cara sederhana pada proses *crossover* yaitu dengan memilih satu titik yang dipisahkan secara acak dan kemudian membentuk offspring dengan mengkombinasikan segmen dari satu induk ke sebelah kiri dari titik yang dipisahkan dengan segmen dari induk yang lain ke sebelah kanan dari titik yang dipisahkan.

b. Mutation (Mutasi)

Proses mmerupakan untuk mengubah salah satu atau lebih beberapa gen dari suatu kromosom. Proses ini berfungsi untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi.

F. Parameter Kontrol

Parameter kontrol genetika diperlukan untuk mengendalikan operator-operator seleksi. Pemilihan parameter genetika menentukan penampilan kinerja algoritma genetika dalam memecahkan masalah (Desiani dkk, 2006). Ada dua parameter dasar dari algoritma genetika, yaitu probabilitas *crossover* (P_c) dan probabilitas mutasi (P_m).

a. Probabilitas *Crossover* (P_c)

Probabilitas *crossover* akan mengendalikan operator *crossover* dalam setiap generasi dalam populasi yang mengalami *crossover*. Semakin besar nilai probabilitas *crossover*, akan semakin cepat struktur individu baru terbentuk ke dalam populasi. Apabila nilai probabilitas *crossover* terlalu besar, maka individu yang merupakan kandidat solusi terbaik mungkin akan dapat hilang lebih cepat pada generasi selanjutnya. Nilai probabilitas *crossover* yang disarankan adalah berkisar antara 80 % - 95 %.

b. Probabilitas mutasi (P_m)

Probabilitas mutasi akan mengendalikan operator mutasi pada setiap generasi dengan peluang mutasi yang digunakan lebih kecil daripada peluang *crossover*. Pada seleksi alam murni, mutasi jarang sekali muncul, sehingga operator mutasi pada algoritma genetik tidak selalu terjadi. Nilai probabilitas mutasi yang disarankan kecil antara 0.5% - 1%.

2.2.5 Tabu Search

Tabu Search adalah *local search* meta heuristik yang dapat digunakan untuk memecahkan masalah optimasi kombinatorial. Jika dibandingkan dengan teknik *local search hill climbing* atau pencarian *local hill climbing*, itu kurang

cenderung ke dalam solusi *local optim*. *Tabu search* memiliki kecepatan eksekusi lebih tinggi dari pada algoritma genetika karena *tabu search* tidak kembali pada solusi yang sudah dieksplorasi, mengingat bahwa adalah tabu. Hal ini dimungkinkan karena masing-masing *tabu search* yang bergerak dicatat untuk mencegah *revisit* atau pengulangan evaluasi pada solusi yang sudah ada [1]. Solusi yang sudah ada sebelumnya dicegah dengan menggunakan memori yang disebut dengan Tabu List. Tabu List yang ada pada *tabu search* digunakan untuk menyimpan sekumpulan solusi yang baru saja dievaluasi. Selama proses optimasi, pada setiap iterasi, solusi yang akan dievaluasi akan dicocokkan terlebih dahulu dengan isi tabu list. Apabila solusi tersebut sudah ada pada tabu list, maka solusi tersebut tidak akan dievaluasi lagi pada iterasi berikutnya. Apabila sudah tidak ada lagi solusi yang tidak menjadi anggota tabu list, maka nilai terbaik yang baru saja diperoleh merupakan solusi yang sebenarnya [3]. Berikut adalah tahap – tahap penerapan algoritma *tabu search*

Pengembangan Algoritma Berbasis Tabu Search

Pengembangan algoritma dilakukan sesuai dengan tahap Tabu Search, yaitu tahap pencarian awal, tahap intensifikasi dan tahap diversifikasi [15] .

Tahap Pencarian Awal

Tahap pencarian awal (*preliminary search*) dilakukan dengan langkah-langkah berikut.

Langkah 0 : Inisialisasi

Pertama, masukkan input. Input yang diperlukan adalah kendala presedensi, waktu siklus yang diinginkan dan waktu proses masing-masing operasi. Kedua, tentukan nilai dari parameter yang digunakan. Berikut adalah parameter-parameter yang harus ditentukan nilainya terlebih dahulu: (1) Tabu list size (T_s), yaitu parameter yang mengatur berapa lama dan berapa jumlah move dalam kondisi tabu. (2) Iterasi maks (N_{maks}), yaitu parameter yang mengatur banyaknya

iterasi maksimum yang dikehendaki. Ketiga, tentukan solusi awal.

Langkah 1 : Pembentukan Candidate Move

Pembentukan candidate move dilakukan dengan langkah-langkah berikut. Pertama, bangkitkan candidate move dengan metode neighborhood search. Proses pencarian candidate move ini dilakukan sampai jumlah candidate move yang diinginkan telah diperoleh (jmlhmaks). Dalam proses pencarian candidate move, solusi dapat dihasilkan dengan menggunakan proses swap maupun insert. Proses ini masing-masing memiliki probabilitas yang sama untuk menghasilkan sebuah solusi. Jika penggunaan metode yang dipilih tidak menghasilkan solusi, maka proses dilanjutkan dengan metode lainnya. Sebagai contoh jika proses pencarian dengan menggunakan metode insert tidak dapat menghasilkan solusi, maka proses pencarian akan diganti dengan menggunakan metode swap, begitupun sebaliknya

Langkah 2 : Pemilihan Move Terbaik

Pada langkah ini akan dipilih move terbaik dari beberapa candidate move yang telah diperoleh. Namun pertama-tama perlu dicek terlebih dahulu apakah semua move dari setiap candidate move yang ada berstatus tabu atau tidak. Jika semua move berstatus tabu maka pemilihan candidate move terbaik akan didasari pada kriteria aspirasi. Jika tidak semua move berstatus tabu maka pilih candidate move terbaik yang diperoleh berdasarkan hasil evaluasi candidate move pada langkah 1.

Langkah 3 : Kriteria Aspirasi

Kriteria aspirasi merupakan sebuah kriteria yang mengizinkan tetap dilakukannya sebuah move yang berstatus tabu.

Tahap Intensifikasi

Tahap kedua dalam algoritma Tabu Search adalah tahap intensifikasi. Pada tahap intensifikasi proses pencarian akan lebih difokuskan pada solusi terbaik yang telah ditemukan. Proses intensifikasi dilakukan secara berulang sampai

mencapai nilai iterasi intensifikasi maksimum (nimaks). Langkah-langkah dalam proses intensifikasi adalah sebagai berikut :

- Langkah 0 : inialisasi: set $S = S^*$, kosongkan tabu list dan set $n_i = 0$, $ntsi = 0$.
- Langkah 1 : pembentukan candidate move.
- Langkah 2 : pilih move terbaik.
- Langkah 3 : kriteria aspirasi.
- Langkah 4 : evaluasi candidate move terpilih.
- Langkah 5 : kriteria pemberhentian.

Tahap Diversifikasi

Tahap ketiga dalam algoritma Tabu Search adalah tahap diversifikasi. Pada tahap diversifikasi, proses pencarian akan lebih meyebar untuk membuka daerah-daerah baru yang belum dikunjungi. Proses diversifikasi ini dilakukan terus hingga tercapai nilai iterasi diversifikasi maksimum (ndmaks).

Langkah-langkah dalam proses diversifikasi adalah sebagai berikut.

- Langkah 0 : inialisasi
- Langkah 1 : pembentukan candidate move.
- Langkah 2 : pilih move terbaik.
- Langkah 3 : kriteria aspirasi.
- Langkah 4 : evaluasi candidate move terpilih.
- Langkah 5 : Kriteria pemberhentian.

2.2.6 Standard Deviasi

Standar deviasi merupakan ukuran keragaman atau variasi data statistik yang biasanya digunakan, standar deviasi juga biasa disebut simpangan baku yang merupakan akar kuadrat dari varian. Tujuan dari penghitungan standar deviasi adalah untuk

mengetahui keragaman kelompok data. Berikut merupakan rumusan dari standar deviasi:

$$s = \sqrt{\frac{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}{n(n-1)}}$$

Dari rumus tersebut, s merupakan standar deviasi (simpangan baku), X_i merupakan nilai x ke i , \bar{x} merupakan rata-rata dari X , dan n merupakan ukuran sampel dimana pada standar deviasi yang memiliki nilai kecil menunjukkan bahwa data-data pada himpunan tersebut mendekati atau berada pada sekitar rata-rata nilai himpunan [16].

2.2.7 Error Variance

Untuk pengukuran tingkat validitas dari perhitungan dibutuhkan suatu indikator yang dapat menunjukkan tingkat kesalahan atau *error* pada suatu model yang sudah ditetapkan, sehingga dalam mengukur tingkat kesalahan digunakan *error variance*. Model dapat dikatakan valid apabila nilai $E \leq 30\%$. Berikut merupakan perumusan dari error variance [17]:

$$E = \frac{|\text{Standard deviasi simulasi} - \text{Standard deviasi data}|}{\text{Standard deviasi data}}$$

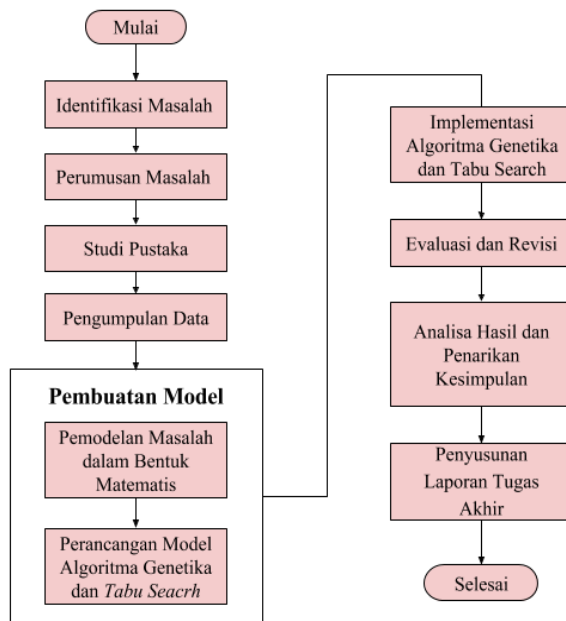
(Halaman ini sengaja dikosongkan)

BAB III METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan langkah langkah pengerjaan yang dilakukan pada penelitian tugas akhir ini. Langkah langkah yang akan dilakukan secara terstruktur agar dapat mempermudah dalam melakukan penelitian tugas akhir.

3.1 Urutan pelaksanaan proses penelitian

Pada bab ini dijelaskan mengenai proses teknik pengumpulan dan analisis data, model pendekatan yang digunakan, rancangan penelitian, ujicoba dan cara evaluasi. Gambar 3.1 merupakan diagram langkah penelitian yang akan dilakukan.



Gambar 3.1 Metode Penelitian

3.1.1 Identifikasi Masalah

Dalam tahap identifikasi masalah, dilakukan studi literature dengan mencari jurnal, paper, tugas akhir atau penelitian-penelitian terkait mengenai permasalahan optimasi alokasi khususnya alokasi sumberdaya sehingga dapat ditentukan metode penyelesaian yang sesuai. Selain itu, pada penelitian tugas akhir ini juga dilakukan identifikasi masalah melalui wawancara terhadap pihak terkait yaitu pada perusahaan untuk mengetahui permasalahan yang terjadi terkait pengalokasian kapal pada PT. XYZ. Pada tahap ini juga dilakukan pencarian materi pendukung lainnya dari website resmi ataupun dokumen kebijakan umum yang membahas mengenai jasa pelayaran dan pengalokasian kapal.

3.1.2 Perumusan Masalah

Dalam memulai melakukan penelitian tugas akhir ini perlu dilakukan perumusan masalah untuk menentukan tujuan yang akan di capai dan faktor faktor yang dapat mempengaruhi serta penggunaan metode penyelesaian yang akan digunakan

3.1.3 Studi Pustaka

Studi pustaka dapat dilakukan dengan cara mengumpulkan referensi, mempelajari dan memahami serta mengutip konsep atau teori dari sejumlah literature seperti buku, penelitian sebelumnya dan dokumen terkait. Pada tahapan ini dapat dilakukan dengan berfokus pada metode yang akan digunakan dalam menyelesaikan masalah yang berkaitan dengan alokasi yaitu metode algoritma genetika.

3.1.4 Pengumpulan Data

Pada tahap ini dilakukan proses pengumpulan data yang dibutuhkan sebagai masukan dalam penyelesaian tugas akhir ini. Data di ambil dari perusahaan penyedia jasa pelayaran yaitu salah satu perusahaan pelayaran di Surabaya. Metode pengambilan data yang digunakan yaitu metode pengumpulan bahan dokumen dan wawancara. Pengambilan data dilakukan

dengan permohonan kepada perusahaan dengan keterangan person in charge (PIC) sebagai pendukung.

3.1.5 Pemodelan Masalah dalam Bentuk Matematis

Setelah melakukan studi pustaka, selanjutnya yaitu membuat model matematis dari permasalahan yang ada. Dalam memodelkan permasalahan, dilakukan penerjemahan permasalahan yang ada kedalam bentuk matematis agar dapat diselesaikan dengan cara matematis. Pada tahap ini ditentukan nilai awal gen – gen sesuai dengan batasan yang ditentukan. Sekumpulan Gen – gen yang terbentuk akan menjadi satu kromosom yang mewakili satu solusi. Sekumpulan kromosom dapat membentuk sebuah individu, dimana sekumpulan individu yang di proses bersama dalam satu siklus proses evolusi dapat membentuk populasi awal. Dalam tugas akhir ini kromosom mewakili muatan yang dibawa kapal i menuju tujuan j pada perjalanan ke k. Permasalahan yang digunakan adalah pengalokasian kapal barang yang akan dijabarkan menjadi model matematis seperti penjelasan sebelumnya yang dapat dioperasikan dalam pengerjaan menggunakan algoritma genetika dan *tabu search*, sehingga dapat menemukan solusi yang optimum.

3.1.6 Perancangan Model Algoritma Genetika dan *Tabu Search*

Paada tahap ini dilakukan perancangan program komputer algoritma genetika dan *tabu search* berdasarkan model matematis yang telah ditentukan pada tahapan sebelumnya. Selanjutnya dilakukan kombinasi metode algoritma genetika dan *tabu search* untuk menemukan hasil yang optimum.

3.1.7 Implementasi Algoritma Genetika dan Tabu Search

Pada tahap ini akan dilakukan implementasi algoritma genetika dan *tabu search* dalam pengalokasian kapal peti kemas berdasarkan perancangan model algoritma genetika yang telah dibuat pada tahap sebelumnya.

3.1.8 Evaluasi dan Revisi

Pada tahap ini akan dilakukan evaluasi dari tahapan pengerjaan sebelumnya. Evaluasi ini bertujuan untuk melakukan perbaikan pada bagian yang belum benar. Setelah ditemukan kesalahan atau terdapat bagian yang belum benar maka akan dilakukan perbaikan atau yang dapat disebut sebagai revisi. Sehingga diharapkan dengan adanya tahapan ini, pengerjaan tugas akhir dapat diselesaikan dengan sempurna.

3.1.9 Analisa Hasil dan Penarikan Kesimpulan

Setelah melakukan semua tahapan di atas, dilakukan perbandingan hasil optimasi, yaitu antara kondisi sebelum dilakukan optimasi dan setelah dilakukan optimasi. Sehingga dapat ditentukan apakah hasil tugas akhir sesuai dengan yang diharapkan. Selain itu dapat diambil suatu kesimpulan untuk dijadikan saran dan rekomendasi kepada perusahaan terkait.

3.1.10 Penyusunan Laporan Tugas Akhir

Setelah melakukan semua tahapan di atas, tahapan terakhir adalah tahap penyusunan laporan tugas akhir sebagai dokumentasi dari hasil pengerjaan penelitian tugas akhir. Semua kegiatan yang dilakukan dalam penelitian tugas akhir akan didokumentasikan ke dalam sebuah buku tugas akhir sesuai dengan format yang telah ditentukan.

3.2 Bahan dan peralatan yang digunakan

Pada sub bab ini akan dijelaskan bahan dan peralatan yang akan digunakan dalam pengerjaan tugas akhir. Bahan dimaksud adalah seperti material dan data. Sedangkan peralatan yang dimaksud adalah seperti alat-alat uji coba, perangkat keras atau perangkat lunak.

3.2.1 Bahan yang Digunakan

Pada tugas akhir ini digunakan bahan yaitu berupa data yang diperoleh dari perusahaan. Data tersebut diperoleh dalam

format file .xls yang berisi data-data mengenai proses pengiriman barang berupa *container* menggunakan kapal.

3.2.2 Alat yang Digunakan

Pada tugas akhir ini digunakan alat berupa perangkat lunak MATLAB R2013a sebagai tools untuk menyelesaikan permasalahan menggunakan algoritma genetika dan *tabu search*, perangkat lunak Windows 8 dan Windows 10. Selain menggunakan perangkat lunak, juga digunakan alat berupa perangkat keras komputer dan laptop yang digunakan sebagai alat untuk membantu dalam menyelesaikan tugas akhir ini.

(Halaman ini sengaja dikosongkan)

BAB IV PERANCANGAN

Bab ini menjelaskan mengenai perancangan model matematis dari permasalahan yang ada dan perancangan model dalam bentuk kombinasi algoritma genetika dan *tabu search*. Sebelum memulai proses optimasi menggunakan kombinasi algoritma genetika dan *tabu search*, harus dilakukan penentuan parameter variabel yang terlibat terlebih dahulu. Parameter yang telah ditentukan akan digunakan pada tahap selanjutnya dalam penyelesaian permasalahan optimasi menggunakan metode algoritma genetika dan *tabu search*.

4.1 Pemodelan Permasalahan dalam Bentuk Matematis

Sebelum memulai perancangan model kombinasi algoritma genetika dan *tabu search*, terlebih dahulu dilakukan pemodelan matematis dari permasalahan atau studi kasus yang ada. Pada tugas akhir digunakan studi kasus pada perusahaan pelayaran yaitu mengenai alokasi kapal peti kemas dengan tujuan memaksimalkan *profit* dan memaksimalkan muatan kapal. Muatan kapal yang dimaksud pada tugas akhir ini berupa satuan *container*.

4.1.1 Deskripsi Data

Pada bagian ini akan dijelaskan data-data yang akan digunakan dalam pengerjaan tugas akhir. Data yang digunakan adalah data perusahaan yang akan di olah yang kemudian menjadi masukan pada MATLAB. Data – data perusahaan yang digunakan dalam penyelesaian tugas akhir ini adalah data yang berkaitan dengan proses pengiriman barang menggunakan kapal peti kemas dari mulai data nama kapal, data kapasitas kapal, data permintaan (*demand*), data pelabuhan tujuan, dan data tarif atau harga sewa yang diberikan perusahaan untuk masing-masing kontainer. Tabel 4.1 adalah gambaran data yang akan di olah, dengan kata lain data berikut bukan data yang akan langsung menjadi masukan pada program tetapi melalui pengolahan terlebih dahulu. Data

yang di tampilkan pada tabel di bawah ini adalah contoh sebagian kecil dari data yang ada. Dari empat pelabuhan tujuan, di ambil data dengan tujuan makassar pada bula Januari 2011 sebagai contoh.

Tabel 4.1 Deskripsi Data

Surabaya - Makassar - Surabaya						
No	Nama Kapal	Kapasitas (teus)	Muatan (teus)	Tarif Sewa Container (Rp juta)	Pendapatan (Rp juta)	Load Factor
1	KM Segoro Mas (SBY-KMS-AB)	296	80	1,8	144	27%
2	KM Cahya Mas	650	350	1,8	630	54%
3	KM Cahya Mas	650	350	1,8	630	54%
4	KM Cahya Mas	650	350	1,8	630	54%
5	KM Cahya Mas	650	350	1,8	630	54%
6	KM Cahya Mas	650	350	1,8	630	54%
7	KM Cahya Mas	650	350	1,8	630	54%

4.1.2 Penentuan Variabel

Dalam pembuatan model matematis sebelum menentukan fungsi *fitness* perlu ditentukan terlebih dahulu variabel yang terlibat dalam proses penyelesaian algoritma genetika dan *tabu search*. Variabel yang terlibat di peroleh berdasarkan faktor-faktor yang mempengaruhi dan solusi yang ingin di capai. Berdasarkan studi kasus terdapat variabel permintaan (*demand*), variabel tarif kontainer, dan variabel kapasitas muat kapal yang nantinya akan menjadi variabel input pada MATLAB. Selain variabel input, juga terdapat variabel keputusan yang mewakili nilai satu gen yang akan membentuk kromosom. Variabel tersebut adalah muatan kontainer yang akan di bawa oleh kapal i ke tujuan j pada perjalanan ke k dan variable keuntungan yang diperoleh perusahaan dalam proses pengiriman barang untuk memenuhi permintaan pada tujuan tertentu. Berikut adalah variable – variable yang terlibat :

- Variabel Input

$demand$ = jumlah permintaan *container* yang harus dikirim menggunakan kapal i ke tujuan j.

P_{ij} = tarif/harga sewa perkontainer kapal i ke tujuan j

Cap_i = kapasitas muat kapal i

- Variabel Output

X_{ijk} = jumlah *container* yang dikirim menggunakan kapal i ke tujuan j pada perjalanan ke-k.

Bestprofit = keuntungan maksimal perusahaan berdasarkan kromosom terpilih

4.1.3 Perumusan Fungsi *Fitness*

Fungsi *fitness* pada tugas ahir ini adalah untuk memaksimalkan *profit* dengan melakukan perkalian jumlah kontainer yang dikirim menggunakan kapal *i* ke tujuan *j* pada perjalanan ke-*k* dengan harga sewa perkontainer kapal *i* ke tujuan *j*, dan memaksimalkan muatan kapal dengan meminimalkan kapasitas kapal yang kosong. Sehingga untuk fungsi *fitness* yaitu memaksimalkan *profit* dan meminimalkan *opportunity cost*. *Opportunity cost* diperoleh dari jumlah kapasitas kapal yang kosong atau tidak terisi kontainer dikalikan dengan biaya sewa perkontainer. Sehingga dengan meminimalkan *opportunity cost* maka dapat meminimalkan kapasitas kapal yang kosong dan memaksimalkan muatan. Berikut adalah fungsi *fitness* dalam bentuk matematis.

$$\text{Max } Z = \sum X_{ijk} \times P_{ij} - \sum (C_{api} - X_{ijk}) \times P_{ij}$$

4.1.4 Perumusan Batasan Model

Setelah menentukan variable – variable yang terkait selanjutnya perlu menentukan batasan yang mempengaruhi permasalahan. Batasan dibuat untuk membatasi fungsi *fitness* dan sebagai kriteria penghentian yang digunakan pada proses perulangan. Batasan yang digunakan pada tugas akhir diantaranya total jumlah muatan kontainer yang dikirim harus sama dengan jumlah permintaan pada masing-masing tujuan. muatan *container* yang dibawa tidak boleh melebihi kapasitas kapal, dan yang terakhir digunakan batasan berapakah frekuensi kapal dapat melakukan pengiriman pada masing-masing tujuan. Berikut adalah batasan yang dipertimbangkan dalam permasalahan.

a. Batasan Total Muatan yang di Kirim

Batasan ini digunakan untuk membatasi total kontainer yang dikirim ke masing-masing tujuan agar sesuai dengan permintaan. Total kontainer yang dikirimkan oleh semua

kapal yang menuju tujuan j tidak boleh lebih atau kurang dari jumlah permintaan pada tujuan j .

- Jika total muatan *container* yang ada pada lokasi j kurang dari permintaan, maka kapal selanjutnya yang melakukan pengiriman ke tujuan tersebut akan membawa muatan dari hasil random dengan batas bawah 0 sampai batas kapasitas maksimal kapal.

Jika

$$\sum X_j < demand$$

maka

$$X_{ijk} = \sum_0^{cap i} rand i$$

- Jika hasil random tersebut di atas ditambah akumulasi total *container* yang sudah ada pada lokasi j lebih dari permintaan, maka kapal yang akan mengirimkan barang menuju lokasi tersebut akan membawa muatan berjumlah selisih antara jumlah permintaan dan total *container* yang sudah berada pada lokasi j .

Jika

$$\sum_0^{cap i} rand i + \sum X_j > demand$$

Maka

$$X_{ijk} = demand - X_j$$

- Jika total barang yang ada di lokasi j sama dengan permintaan atau *demand* maka muatan kapal akan

diberi nilai sama dengan 0 karena permintaan sudah terpenuhi.

Jika

$$X_j = demand$$

Maka

$$X_{ijk} = 0$$

b. Batasan Jumlah Muatan Kapal

Batasan ini digunakan untuk membatasi jumlah muatan yang di bawa masing-masing kapal i menuju tujuan j tidak boleh melebihi kapasitas maksimum kapal. Sehingga dapat di tuliskan dalam bentuk matematis berikut :

$$X_{ijk} \leq Cap\ i$$

c. Batasan Berapa Kali Kapal Dapat Melakukan Pengiriman

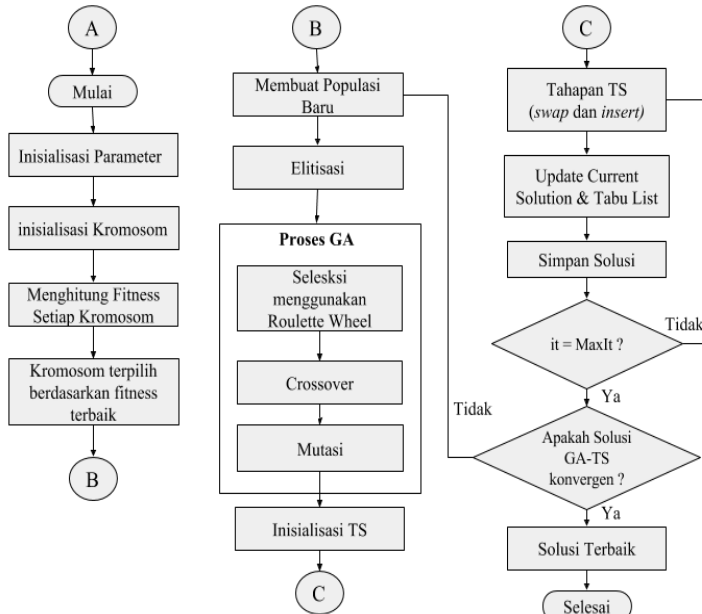
Batasan ini digunakan untuk membatasi jumlah atau frekuensi kapal i dalam melakukan pengiriman ke tujuan j . Sehingga tidak akan terjadi pengiriman dengan kapal yang sama terus menerus. Kapal i dapat melakukan pengiriman ke tujuan j maksimal sebanyak BB . Dimana BB dihasilkan dari perhitungan jumlah permintaan tujuan j dibagi dengan kapasitas maksimal kapal i . Berikut adalah bentuk matematisnya :

$$BB = \frac{Permintaan\ perbulan}{Cap\ i}$$

4.2 Perancangan Algoritma Genetika dan *Tabu Search*

Setelah memodelkan permasalahan dalam bentuk matematis pada sub bab sebelumnya, selanjutnya melakukan perancangan model algoritma genetika dan *tabu search*. Pada tahap desain algoritma genetika dan *tabu search* , akan di lakukan proses penyelesaian dengan menggunakan algoritma genetika terlebih

dahulu baru kemudian metode *tabu search* diterapkan. Berdasarkan studi kasus pada tugas akhir ini dilakukan implementasi menggunakan algoritma genetika dengan nilai kontinyus. Gambar 4.1 adalah *flowchart* atau diagram alir proses algoritma genetika dan *tabu search*.



Gambar 4.1 Flowchart GA-TS

4.2.1 Inisialisasi Parameter

Pada subbab ini akan membahas mengenai proses penentuan parameter variabel yang akan digunakan dan berpengaruh dalam proses penerapan metode algoritma genetika dan *tabu search*. Terdapat parameter dasar dalam algoritma genetika yang perlu ditentukan yaitu probabilitas kawin silang (*crossover*), probabilitas pindah silang (mutasi). Variabel – variabel tersebut akan menentukan berapa banyak individu yang akan dihasilkan, berapa banyak terjadinya proses kawin silang dan mutasi. Berikut adalah proses penentuan nilai dari variabel yang akan ditentukan di atas.

a. Probabilitas Kawin Silang (*Crossover*)

Probabilitas *crossover* (P_c) adalah parameter berupa angka dalam bentuk presentase yang akan mempengaruhi seberapa banyak terjadinya proses persilangan antara sesama individu. Probabilitas *crossover* akan mempengaruhi jumlah kromosom yang akan mengalami *crossover* dan mengendalikan operator *crossover* dalam setiap generasi pada populasi yang mengalami *crossover*. Semakin besar nilai probabilitas *crossover*, akan semakin cepat struktur individu baru terbentuk ke dalam populasi. Apabila nilai probabilitas *crossover* terlalu besar, maka individu yang merupakan kandidat solusi terbaik mungkin akan dapat hilang lebih cepat pada generasi selanjutnya. Nilai probabilitas *crossover* yang disarankan adalah berkisar antara 80 % - 95 % [12].

b. Probabilitas Mutasi

Probabilitas mutasi (P_m) adalah parameter berupa angka dalam bentuk presentase yang akan mempengaruhi seberapa banyak terjadinya mutasi / perpindahan nilai dalam suatu populasi. Probabilitas mutasi akan mengendalikan operator mutasi pada setiap generasi dengan peluang mutasi yang digunakan lebih kecil daripada peluang *crossover*. Pada seleksi alam murni, mutasi jarang sekali muncul, sehingga operator mutasi pada algoritma genetik tidak selalu terjadi. Nilai probabilitas mutasi yang disarankan antara 0.5% - 1% [12].

c. Jumlah individu

Dalam tahapan algoritma genetika terdapat proses inisialisasi atau pembentukan populasi awal, dimana akan menghasilkan sejumlah atau sekumpulan individu. Sehingga perlu ditentukan jumlah individu yang harus dibentuk dalam satu populasi. Jumlah individu menunjukkan jumlah kromosom yang terdapat dalam populasi (dalam satu generasi). Jika hanya sedikit

kromosom dalam populasi maka algoritma genetika akan mempunyai sedikit variasi kemungkinan untuk melakukan crossover antara orangtua karena hanya sebagian kecil dari search space yang dipakai. Sebaliknya jika terlalu banyak maka algoritma genetika akan berjalan lambat [18]. Pada tugas akhir ini jumlah individu yang akan digunakan tergantung dari nilai N yang akan di input oleh user.

d. Jumlah Generasi

Pada penyelesaian algoritma genetika perlu ditentukan jumlah generasi atau jumlah iterasi yang akan berpengaruh pada solusi yang dihasilkan. Pada tugas akhir ini jumlah iterasi dapat ditentukan oleh user dengan mengganti nilai variabel maxit sesuai dengan yang diinginkan.

4.2.2 Bangkitkan Populasi

Pada implementasi algoritma genetika kontinyus tidak menggunakan proses encoding-decoding [14]. Masing-masing kromosom dibangkitkan dengan menginputkan nilai sebanyak ukuran populasi secara random pada MATLAB dikodekan dengan N. Populasi dibuat dengan menentukan jumlah gen dalam satu kromosom yang nantinya beberapa kromosom terbentuk menjadi individu dan kemudian menjadi populasi. Sebelum membangkitkan populasi akan dibuat header terlebih dahulu untuk menunjukkan kapal i, tujuan j pada perjalanan k. Kemudian dilanjutkan dengan membangkitkan populasi dengan menentukan kromosom yang akan di evaluasi pada tahap selanjutnya. Dilakukan random untuk menentukan nilai kromosom yang kemudian akan terbentuk suatu populasi. Banyaknya kromosom dalam satu populasi ditentukan berdasarkan nilai N yang di input di awal.

- **Gen**

Pada kasus ini dapat dilihat pada Gambar 4.2, gen merupakan perwakilan nilai dari jumlah muatan kontainer yang harus di bawa oleh kapal i menuju tujuan j pada

perjalanan k. Jumlah gen dalam satu kromosom adalah sebanyak panjang populasi yang terbentuk berdasarkan kombinasi vektor yang terjadi antara variabel jumlah kapal, jumlah tujuan dan jumlah perjalanan atau variabel bb yaitu sebanyak 195 gen.



Gambar 4.2 Gen Bagian dari Kromosom

- **Individu/Kromosom**

Pada kasus ini, suatu individu merupakan kromosom yang berisi solusi muatan kontainer yang akan di bawa oleh kapal i ke tujuan j pada perjalanan ke k. Kromosom terdiri dari sekumpulan gen, yang berisi solusi muatan yang harus dibawa oleh kapal yang telah ditentukan menuju tujuan Makassar, Bitung, Ambon dan Banjarmasin. Gambar 4.3 merupakan potongan kromosom yang terdiri dari 11 gen dari jumlah keseluruhan gen yaitu 195. Baris pertama menunjukkan kapal apa yang digunakan, baris kedua menunjukkan pelabuhan tujuan, baris ketiga digunakan sebagai simbol pembeda kapal yang melakukan perjalanan pada lokasi yang sama, dan baris keempat menunjukkan jumlah muatan yang dibawa.

Kotak berwarna merah pada Gambar 4.3 menunjukkan nilai satu gen yang menjelaskan bahwa untuk mengirim muatan sebanyak 473 menuju pelabuhan 1 akan digunakan kapal 8. Huruf “a” pada kotak berwarna merah digunakan sebagai simbol bahwa kapal 8 yang digunakan untuk membawa muatan 473 menuju lokasi 1 berbeda perjalanan dengan kapal 8 yang membawa muatan 535 pada tujuan 1.

1	7	8	8	10	10	10	11	1	2	7
1	1	1	1	1	1	1	1	2	2	2
a	a	a	b	a	b	c	a	a	a	a
149	23	473	535	262	269	283	186	650	296	538

Gambar 4.3 Potongan Kromosom

- **Populasi**

Pada kasus ini, suatu populasi merupakan sekumpulan solusi dari muatan kontainer yang akan di bawa oleh kapal i ke tujuan j pada perjalanan ke k. Dengan kata lain, populasi merupakan sekumpulan dari kromosom. Gambar 4.4 merupakan populasi yang terdiri dari sepuluh kromosom.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	393	0	68	446	0
0	0	0	0	0	35	199	0	0	0	0	0	0	0	0	0	0	0	0	469
0	428	0	0	0	0	0	12	0	0	0	0	0	0	218	0	0	0	128	0
0	0	0	0	0	0	0	0	0	0	251	0	362	0	0	0	0	0	0	0
0	0	0	466	0	0	0	0	0	0	0	0	0	354	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	256	0	293	0	0	129	0	0	0	0	406
0	492	0	0	0	0	36	0	0	0	0	0	6	0	0	0	0	0	0	0
453	0	0	0	0	0	0	0	0	0	103	0	0	0	0	0	0	0	0	0
0	632	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	271
0	575	0	0	0	0	0	0	0	0	0	0	227	0	19	142	284	0	0	0

Gambar 4.4 Populasi

4.2.3 Evaluasi Populasi

Masing-masing solusi atau kromosom dalam populasi dievaluasi dengan memasukkan ke fungsi *fitness* [14]. Pada tugas akhir ini digunakan fungsi *fitness* yaitu memaksimalkan *profit* perusahaan penyedia jasa pelayaran. Untuk melakukan evaluasi, dilakukan dengan menghitung nilai *fitness* untuk setiap kromosom berdasarkan jumlah muatan kontainer yang dibawa di kalikan dengan biaya sewa perkontainer dikurangkan dengan *opportunity cost*. Seperti yang sudah dijelaskan pada bab 4.1 sub bab 4.1.3 mengenai perumusan fungsi *fitness*.

4.2.4 Elitisme

Elitisme dalam algoritma genetika berarti usaha untuk mempertahankan individu-individu terbaik yang telah diperoleh di suatu generasi ke dalam generasi selanjutnya. Sehingga individu-individu terbaik akan tetap muncul di populasi berikutnya [14]. Pada tugas akhir ini dilakukan penyalinan individu terbaik untuk dapat mempertahankan individu tersebut dengan melakukan pemilihan kromosom terbaik. Kromosom yang terpilih yaitu kromosom dengan nilai tertinggi untuk disalin sebanyak 4 kali jika ukuran populasi berjumlah genap, dan disalin sebanyak 3 kali jika ukuran populasi berjumlah ganjil.

4.2.5 Seleksi Kromosom

Seleksi dilakukan dengan memilih kromosom yang mempunyai nilai fitness tertinggi. Pada tugas akhir ini dilakukan seleksi dengan menggunakan metode *Roulette Wheel*. Untuk memulai menggunakan *Roulette Wheel* sebelumnya lakukan proses perhitungan kumulatif nilai fitness terlebih dahulu [14]. Pada tugas akhir ini dilakukan perhitungan kumulatif dari nilai fitness yang kemudian hasilnya akan dibagi dengan total kumulatif. Selanjutnya dilakukan seleksi dengan menggunakan roda lotere yang nantinya akan menghasilkan *parent 1* dan *parent 2*. Orangtua atau *parent* yang dihasilkan tersebut merupakan kromosom terbaik dari hasil proses *roulette wheel* yang nantinya akan dilakukan proses *crossover*.

4.2.6 Crossover

Setelah terpilih kromosom dari proses seleksi sebagai induk, selanjutnya dilakukan *crossover* atau kawin silang [14]. Pada tugas akhir ini, proses *crossover* dilakukan dengan membangkitkan nilai random terlebih dahulu. Jika nilai random kurang dari nilai probabilitas *crossover* maka akan terjadi proses *crossover*. Jika tidak maka tidak terjadi proses *crossover*. Pada proses *crossover* terjadi kawin silang antara *parent 1* dan *parent 2* yang kemudian akan membentuk

keturunan baru. Jika tidak terjadi proses *crossover* maka keturunan 1 dan keturunan 2 akan tetap berisi nilai *parent 1* dan *parent 2*.

4.2.7 Mutasi

Mutasi dilakukan dengan cara memilih salah satu atau lebih kromosom mana yang akan mengalami mutasi secara acak. Kromosom ini kemudian di isi dengan nilai baru secara acak. Pemilihan kromosom yang mengalami mutasi dipengaruhi oleh parameter probabilitas mutasi [14].

4.2.8 Penerapan *Tabu Search*

Pada tugas akhir ini *tabu search* diterapkan setelah melakukan proses algoritma genetika. Tujuannya adalah untuk mencari solusi yang lebih optimal dan terjadi peningkatan setelah diterapkannya metode *tabu search*. Kromosom masukan pada proses *tabu search* merupakan kromosom yang memiliki nilai fitness terbesar dari semua kromosom dari hasil seluruh generasi pada proses algoritma genetika. Algoritma *tabu search* menggunakan *tabu list* untuk menyimpan sekumpulan solusi yang baru saja dievaluasi. Berikut adalah tahap penerapan metode *tabu search* :

Tahap 1 : melakukan penentuan parameter MTLN (*Maximum Tabu List Size*) dan jumlah maksimum iterasi[1]. Pada tugas akhir dilakukan tahap penentuan parameter dengan menentukan panjang *tabu list* yang akan mengatur jumlah *move* dan melakukan inisialisasi maksimal iterasi dimana maksimal iterasi dapat berubah sesuai dengan jumlah iterasi yang ingin dilakukan.

Tahap 2 : pada tahap ini dilakukan proses inisialisasi solusi awal dan insialisasi *tabu counter* [15]. Pada tugas akhir solusi awal yang dibentuk merupakan solusi yang sama pada proses algoritma genetika yaitu muatan *container* kapal i ke tujuan j pada perjalanan ke k, bentuk header yang menunjukkan kapal apa, tujuan mana, perjalanan ke berapa dan yang terakhir solusi *profit* yang didapat.

Tahap 3 : menghasilkan solusi terbaik dengan proses pembentukan *candidate move*, pada proses ini solusi dapat dihasilkan dengan melakukan proses *swap* dan *insert* [15]. Solusi terbaik yang dihasilkan pada tugas akhir ini berupa *profit* atau keuntungan yang diperoleh perusahaan. Apabila *profit* yang diperoleh pada solusi yang baru lebih besar dari pada solusi *profit* yang sudah ada dan sudah mencapai maksimal iterasi, maka itu merupakan solusi terbaik yang dipilih.

Tahap 4 : melakukan update solusi dan tabu list dan menyimpan solusi terbaik yang pernah ada.

BAB V IMPLEMENTASI

Pada bab ini berisi proses implementasi metode kombinasi algoritma genetika dan *tabu search* kedalam *tools* MATLAB berdasarkan perancangan yang sudah dibuat pada bab sebelumnya.

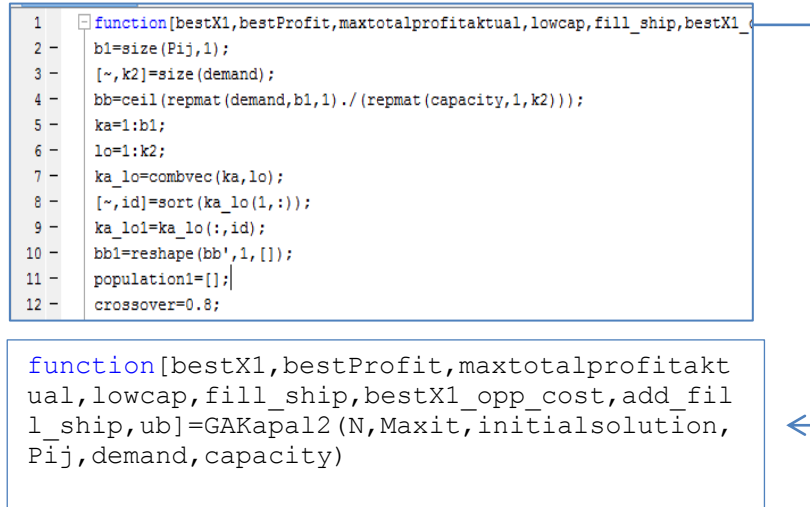
5.1 Inisialisasi parameter

Tahap inisialisasi parameter merupakan tahap penentuan parameter apa saja yang akan digunakan dalam implementasi algoritma genetika pada MATLAB. Seperti pada Gambar 5.1 baris sebelas kode program dapat dilihat penentuan probabilitas *crossover* yaitu 0,8 yang nantinya dapat dilakukan perubahan pada proses uji coba. Selain itu pada kode program dapat dilihat pada baris satu, terdapat sebuah fungsi yang menunjukkan bahwa M-file tersebut bernama “GAKapal2” fungsi tersebut akan menghasilkan variable output *bestX1*, *bestProfit*, *maxtotalprofitaktual*, *lowcap*, *fill_ship*, *bestX1_opp_cost*, *add_fill_ship*, dan *ub*. Variable tersebut berisi nilai-nilai yang akan muncul pada workspace ketika program dijalankan.

Variable output utama yang diharuskan muncul adalah variable *bestX1* dan *bestProfit*. Karena kedua variable ini berisi nilai solusi yang ingin di peroleh. Untuk *bestX1* berisi individu atau kromosom terpilih dimana kromosom tersebut berisi solusi jumlah muatan *container* yang harus di bawa kapal i ke tujuan j dan variable *bestProfit* berisi keuntungan atau *profit* optimal yang di peroleh berdasarkan individu atau kromosom terpilih.

Untuk dapat menjalankan fungsi tersebut dibutuhkan variable input yang harus di inputkan oleh user. Variable input yang dibutuhkan yaitu *N*, *Maxit*, *initialsolution*, *Pij*, *demand*, dan *capacity*. Untuk variable input *Pij*, *demand* dan *capacity* dilakukan *import* dari data perusahaan yang telah diolah dalam

format .xls. Untuk baris kedua hingga sepuluh akan digunakan untuk membuat sebuah header yang akan menunjukkan kapal apa, tujuan mana dan pada perjalanan ke berapa yang akan dijelaskan lebih lanjut pada Gambar 5.1 untuk melihat lebih jelas kode program baris pertama, dapat di lihat perbesaran berikut sesuai dengan panah.



Gambar 5.1 Inisialisasi Parameter

Pada kode program Gambar 5.2 merupakan tahap untuk membentuk sebuah *header* pada populasi untuk menunjukkan kapal apa yang sedang digunakan, menuju tujuan mana dan pada perjalanan keberapa. *Header* tersebut dibuat dengan tujuan untuk menunjukkan solusi yang akan di ambil dan menentukan kapal mana saja yang akan melakukan pengiriman pada tujuan satu yaitu pelabuhan Makasar, tujuan dua pelabuhan Bitung, tujuan tiga pelabuhan Ambon dan tujuan empat yaitu pelabuhan Banjarmasin.


```

13
14 - for p=1:length(ka_lo1)
15 -     poptemp=combvec(ka_lo1(:,p), (1:bb1(:,p)));
16 -     population1=[population1,poptemp];
17 - end
18
19 -     [~,idg]=sort(population1(2,:));
20 -     population=population1(:,idg);
21
22 - for q=1:length(population)
23 -     ub(1,q)=capacity(population(1,q));
24 -     Pijt(1,q)=Pij(population(1,q));
25 - end
26

```

Gambar 5.2 Pembentukan Header

Gambar 5.3 adalah hasil kode program yang dibuat di atas. Dapat dilihat pada kolom satu menjelaskan bahwa kapal 1 menuju tujuan 1 pada perjalanan ke 1. Begitu pula pada kolom delapan menunjukkan bahwa kapal 2 menuju tujuan 1 pada perjalanan ke 4.

	1	2	3	4	5	6	7	8	9
1	1	1	1	1	2	2	2	2	2
2	1	1	1	1	1	1	1	1	1
3	1	2	3	4	1	2	3	4	5
4									
5									

Gambar 5.3 Hasil Header

5.2 Bangkitkan Populasi

Pada tahap ini merupakan tahap membangkitkan populasi dengan menentukan jumlah populasi atau jumlah kromosom yang ingin dibangkitkan di awal yaitu dengan melakukan input nilai N. Pada kode program Gambar 5.4 dapat di lihat variable N dimana N yang dimaksud adalah jumlah populasi yang akan dibangkitkan. Setelah menentukan jumlah kromosom yang ingin di bangkitkan, selanjutnya menentukan nilai kromosom secara acak atau *random*. Penentuan nilai acak tersebut

memiliki batasan yang ditunjukkan pada kode program baris 41-48. Pada kode program tersebut dijelaskan bahwa

- Jika total muatan *container* yang ada pada lokasi j kurang dari permintaan, maka kapal selanjutnya yang melakukan pengiriman ke tujuan tersebut akan membawa muatan dari hasil random dengan batas bawah 0 sampai batas kapasitas maksimal kapal.

Jika

$$\sum X_j < demand$$

maka

$$X_{ijk} = \sum_0^{cap i} rand i$$

- Jika hasil random tersebut di atas ditambah akumulasi total *container* yang sudah ada pada lokasi j lebih dari permintaan, maka kapal yang akan mengirimkan barang menuju lokasi tersebut akan membawa muatan berjumlah selisih antara jumlah permintaan dan total *container* yang sudah berada pada lokasi j .

Jika

$$\sum_0^{cap i} rand i + \sum X_j > demand$$

Maka

$$X_{ijk} = demand - X_j$$

- Jika total barang yang ada di lokasi j sama dengan permintaan atau *demand* maka muatan kapal akan diberi nilai sama dengan 0 karena permintaan sudah terpenuhi.

Jika

$$X_j = demand$$

Maka

$$X_{ijk} = 0$$

Sehingga kode program di bawah ini digunakan untuk membangkitkan solusi awal dan juga digunakan untuk membatasi agar pengiriman barang dari semua solusi yang dibangkitkan tidak kurang atau lebih dari permintaan atau *demand* setiap tujuan yang ada.

```

28 - for i=1:N
29 -     population(3+i,:)=zeros(1,q);
30 -     for t=1:k2
31 -         j(t,:)=(population(2,:)==t);
32 -     end
33 -     for asd=1:size(j,1);
34 -         aa=find(j(asd,:));
35 -         bb=rand(1,size(aa,2));
36 -         cc=[aa;bb];
37 -         [~,idh]=sort(cc(2,:));
38 -         dd=cc(:,idh);
39 -         ee=dd(1,:);
40 -         for xy=1:length(ee)
41 -             if sum(population(3+i,population(2,:)==asd),2)<=demand(1,asd);
42 -                 if demand(1,asd)-sum(population(3+i,population(2,:)==asd),2)>ub(:,ee(:,xy));
43 -                     population(3+i,ee(:,xy))=round(ub(:,ee(:,xy)).*rand);
44 -                 else
45 -                     population(3+i,ee(:,xy))=demand(1,asd)-sum(population(3+i,population(2,:)==asd),2);
46 -                 end
47 -             else
48 -                 population(3+i,ee(:,xy))=0;
49 -             end
50 -         end
51 -     end
52 - end

```

Gambar 5.4 Bangkitkan Populasi

Pada kode program Gambar 5.5 dijelaskan bahwa variable “ps” merupakan ukuran populasi yang terbentuk. Pada kode program di lakukan inisialisasi lagi dikarenakan terdapat *header* yang digunakan untuk menunjukkan urutan kapal, tujuan dan perjalanan pada variable “population”. Sehingga ukuran populasi yang ada pada variable “ps” merupakan ukuran populasi awal dikurangi tiga *header*. Sehingga variable “ps” akan bernilai sama dengan “N”. Selanjutnya untuk kode program baris 56 menunjukkan jumlah maksimal iterasi yang akan dilakukan.

```
53 - population=[population;initialsolution];  
54 - ps=size(population,1)-3;  
55  
56 - for generation=1:Maxit
```

Gambar 5.5 Update Populasi

5.3 Evaluasi Populasi

Pada tahap ini dilakukan penghitungan nilai fitness setiap kromosom. Masing-masing solusi atau kromosom dalam populasi dievaluasi dengan memasukkan ke dalam fungsi *fitness*. Kode program Gambar 5.6 menjelaskan perhitungan fungsi *fitness*, dimana fungsi *fitness*nya adalah *profit* atau keuntungan yang diperoleh perusahaan. Kode program tersebut menjelaskan bahwa jika nilai kromosom lebih dari nol maka nilai y =1 jika tidak maka nilai y=0. Nilai y akan berpengaruh pada perhitungan fungsi *fitness*, dikarenakan *profit* yang akan di peroleh merupakan hasil dari perkalian antara muatan yang dibawa oleh kapal, biaya tiap *container* dan nilai y yang dikurangkan dengan *opportunity cost*. *Opportunity cost* diperoleh dari jumlah kapasitas kapal yang kosong atau tidak terisi kontainer dikalikan dengan biaya sewa perkontainer. Pada Gambar 5.6 dilakukan perbesaran untuk memperjelas kode program.

```

59 ~ for i=1:ps
60 ~     for j=1:q
61 ~         if (population(3+i,j)) > 0
62 ~             y(i,j)=1;
63 ~         else
64 ~             y(i,j)=0;
65 ~         end
66 ~         profit(i,j)=(population(3+i,j).*Pij(population(1,j)).*y(i,j)-(ub(1,j)-
67 ~             population(3+i,j)).*Pij(population(1,j)).*y(i,j));
68 ~         totalprofit(i,1)=sum(profit(i,:),2);
69 ~     end

```

```

for i=1:ps
    for j=1:q
        if (population(3+i,j)) > 0
            y(i,j)=1;
        else
            y(i,j)=0;
        end

        profit(i,j)=(population(3+i,j).*Pij(population(1,j)).*y(i,j)-(ub(1,j)-
        population(3+i,j)).*Pij(population(1,j)).*y(i,j));
    end
    totalprofit(i,1)=sum(profit(i,:),2);
end

```

Gambar 5.6 Evaluasi Populasi

Gambar 5.7 merupakan kode program untuk menghitung keuntungan tanpa dikurangi *opportunity cost* yang dinamakan dengan variable “profitaktual” dan “totalprofit actual”. Perhitungan *profit* berikut tidak dijadikan fungsi *fitness*, hanya saja dijadikan sebagai perbandingan untuk *profit* yang belum dan sudah dilakukan pengurangan dengan *opportunity cost*.

```

71 ~ for i=1:ps
72 ~     for j=1:q
73 ~         if (population(3+i,j)) > 0
74 ~             y(i,j)=1;
75 ~         else
76 ~             y(i,j)=0;
77 ~         end
78 ~         profitaktual(i,j)=(population(3+i,j).*Pij(population(1,j)).*y(i,j));
79 ~     end
80 ~     totalprofitaktual(i,1)=sum(profitaktual(i,:),2);
81 ~ end

```

Gambar 5.7 Profit Aktual

Pada kode program Gambar 5.8 tersebut dijelaskan bahwa kromosom yang terpilih adalah kromosom yang memiliki *profit* paling tinggi.

```

83 - opp_cost=profitaktual-profit;
84 - mintotalprofit=min(totalprofit);
85 - [maxtotalprofit,idk]=max(totalprofit);
86 - maxtotalprofitaktual=max(totalprofitaktual);
87 - bestX1=[population(1:3,:);population(3+idk,:)];
88 - idkbest=find(bestX1(4,:));
89 - bestX=bestX1(:,idkbest);
90 - bestProfit=totalprofit(idk);
91 - bestX1_opp_cost=opp_cost(idk,:);

```

Gambar 5.8 Kromosom Terpilih

5.4 Elitisme

Pada tahap elitisasi ini merupakan proses mempertahankan kromosom terbaik agar tidak hilang dan tetap muncul pada populasi selanjutnya. Kode program Gambar 5.9 tersebut menunjukkan bahwa jika populasi berjumlah genap maka kromosom akan di salin sebanyak empat kali. Jika populasi berjumlah ganjil maka kromosom akan di salin sebanyak tiga kali.

```

115 - if mod(ps,2)==0, %The population size is even
116 -     initialiteration = 8; % The best chromosome is copied 4 time(s)
117 -     temp_population(4,:) = population(3+idk,:);
118 -     temp_population(5,:) = population(3+idk,:);
119 -     temp_population(6,:) = population(3+idk,:);
120 -     temp_population(7,:) = population(3+idk,:);
121 -
122 - else %The population size is odd
123 -     initialiteration = 7; % The best chromosome is copied 3 time(s)
124 -     temp_population(4,:) = population(3+idk,:);
125 -     temp_population(5,:) = population(3+idk,:);
126 -     temp_population(6,:) = population(3+idk,:);
127 - end

```

Gambar 5.9 Elitisme

5.5 Seleksi Kromosom

Pada tahap ini di lakukan seleksi koromosom dengan memilih kromosom yang mempunyai nilai fitness tertinggi. Pada tugas akhir ini digunakan metode *roulette wheel* atau roda lotere.

Proses *roulette wheel* dapat dilihat pada kode program Gambar 5.10. Kode program tersebut merupakan M-file yang bernama “lottery” yang menjalankan proses seleksi menggunakan metode *roulette wheel*. Sedangkan pada kode program Gambar 5.11 akan menghasilkan orangtua sebagai kromosom terbaik yang merupakan hasil dari proses *roulette wheel* pada kode program Gambar 5.10. Oarangtua atau *parent* yang dihasilkan tersebut merupakan kromosom terbaik dari hasil proses *roulette wheel* yang nantinya akan di lakukan proses *crossover*.

```

1  function abcd = lottery(N,fitness)
2  -   totalfitness = sum(fitness);
3  -   cumfitness = 0;
4  -   aa = rand;
5  -   ab = 1;
6  -   while ab <= N
7  -       cumfitness = cumfitness + fitness(ab);
8  -       if(cumfitness/totalfitness) > aa
9  -           abcd = ab;
10 -           break;
11 -       end
12 -       ab = ab + 1;
13 -   end
14 -   end

```

Gambar 5.10 Proses Lottery

```

130 -   for r=initialiteration:2:ps,
131 -       for abc=1:2
132 -           parents(abc,1)=lottery(ps,fitness);
133 -       end
134 -       Father = parents(1,1);
135 -       Mother = parents(2,1);

```

Gambar 5.11 Proses Seleksi

5.6 Crossover

Hasil dari proses seleksi pada tahap sebelumnya akan digunakan pada tahap *crossover*. Pada kode program Gambar 5.12 dilakukan proses pembangkitan nilai random terlebih dahulu. Jika nilai random kurang dari probabilitas *crossover* maka proses *crossover* akan terjadi. Maka nilai pada variable

“temp_population” akan berisi kromosom yang sudah mengalami *crossover* atau dapat dikatakan menghasilkan keturunan baru. Jika nilai random lebih besar dari probabilitas *crossover* maka tidak akan terjadi proses *crossover* dan variable “temp_population” akan tetap berisi kromosom yang sama dengan kromosom orangtua atau *parent*.

```

140 -         if (rand<crossover) %If crossover happened
141 -             Child = aritps(population(3+Father,:),population(3+Mother,:));
142 -             temp_population(r,:) = Child(1,:);
143 -             temp_population(r+1,:) = Child(2,:);
144 -         else % If crossover isn't happened
145 -             %New chromosome is got from father and mother without doing cro
146 -             temp_population(r,:) = population(3+Father,:);
147 -             temp_population(r+1,:) = population(3+Mother,:);
148 -         end
149 -     end

```

Gambar 5.12 Proses Crossover

5.7 Mutasi

Pada tahap ini merupakan proses mutasi dimana akan di lakukan dengan cara memilih salah satu atau lebih kromosom mana yang akan mengalami mutasi secara acak. Pemilihan kromosom yang akan mengalami mutasi dipengaruhi oleh probabilitas mutasi. Dapat di lihat pada kode program Gambar 5.13 pada baris pertama menunjukkan probabilitas mutasi yang digunakan dalam proses mutasi. Pada kodingan tersebut digunakan probabilitas mutasi yaitu sebesar 20% dari populasi. Selanjutnya dapat di lihat pada baris kedua kode program menunjukkan bahwa akan dibangkitkan nilai random dari 1 sampai sejumlah populasi pada variabel “mt”. Kromosom yang terpilih berdasarkan probabilitas mutasi kemudian di isi dengan nilai baru secara acak atau nilai random yang dibangkitkan sebanyak jumlah populasi. Untuk baris kode program selanjutnya dilakukan proses seperti pada saat pembangkitan populasi di tahap awal, dimana diterapkan batasan-batasan dalam membangkitkan nilai random pada kromosom.


```

148 ~ for s=1:(0.2*ps)
149 ~     mt=ceil(rand*ps);
150 ~     temp_population(3+mt,:)=zeros(1,q);
151 ~     for asd=1:size(j,1);
152 ~         aa=find(j(asd,:));
153 ~         bb=rand(1,size(aa,2));
154 ~         cc=[aa;bb];
155 ~         [~,idh]=sort(cc(2,:));
156 ~         dd=cc(:,idh);
157 ~         ee=dd(1,:);
158 ~         for xy=1:length(ee)
159 ~             if sum(temp_population(3+mt,temp_population(2,:)==asd),2)<=demand(1,asd);
160 ~                 if demand(1,asd)-sum(temp_population(3+i,temp_population(2,:)==asd),2)>ub(i,ee(:,xy));
161 ~                     temp_population(3+mt,ee(:,xy))=round(ub(:,ee(:,xy)).*rand);
162 ~                 else
163 ~                     temp_population(3+i,ee(:,xy))=demand(1,asd)-sum(temp_population(3+i,temp_population(2,:)==asd),2);
164 ~                 end
165 ~             else
166 ~                 temp_population(3+mt,ee(:,xy))=0;
167 ~             end
168 ~         end
169 ~     end
170 ~ end
171 ~ population = temp_population;
172 ~ end

```

```

for s=1:(0.2*ps)
    mt=ceil(rand*ps);
    temp_population(3+mt,:)=zeros(1,q);
    for asd=1:size(j,1);
        aa=find(j(asd,:));
        bb=rand(1,size(aa,2));
        cc=[aa;bb];
        [~,idh]=sort(cc(2,:));
        dd=cc(:,idh);
        ee=dd(1,:);
        for xy=1:length(ee)
            if
sum(temp_population(3+mt,temp_population(2,:)==asd),
2)<=demand(1,asd);
                if demand(1,asd)-
sum(temp_population(3+i,temp_population(2,:)==asd),2)
>ub(i,ee(:,xy));
                    temp_population(3+mt,ee(:,xy))=round(ub(:,ee(:,xy)).
.*rand);
                else
                    temp_population(3+i,ee(:,xy))=demand(1,asd)-
sum(temp_population(3+i,temp_population(2,:)==asd),2)
);
                end
            else
                temp_population(3+mt,ee(:,xy))=0;
            end
        end
    end
end
population = temp_population;

```

Gambar 5.13 Proses Mutasi

Setelah semua proses dilakukan hingga pada proses mutasi, selanjutnya dilakukan pemilihan gen pada kromosom. Gen yang membawa muatan hampir penuh dan gen muatan yang membawa paling sedikit akan ditampilkan. Gambar 5.14 dapat dilihat, terdapat variable “lowcap” dimana variable ini akan menampilkan gen mana saja yang membawa muatan paling sedikit pada hasil kromosom terbaik dari proses GA. Sedangkan variable “fill_ship” akan menampilkan gen mana saja yang membawa muatan hampir penuh berdasarkan kromosom terbaik yang terpilih dari proses GA. Hasil dari kode program dapat dilihat pada lampiran D.

```
173 - lowcap=find(and(bestX1(4,:) < (ub(1,:)/2), bestX1(4,:) > 0));
174 - fill_ship=find(bestX1(4,:) > (ub(1,:)/2));
175 - end
```

Gambar 5.14 Pemilihan Gen

5.8 Tabu Search

Tabu search diterapkan setelah program algoritma genetika selesai. Hasil dari solusi terbaik algoritma genetika akan menjadi masukan pada proses *tabu search*. Berikut adalah implementasi metode *tabu search* yang akan dikombinasikan dengan metode algoritma genetika pada tugas akhir ini

5.8.1 Inisialisasi Parameter

Pada tugas akhir dilakukan tahap penentuan parameter dengan menentukan panjang tabu list yang akan mengatur jumlah *move* dan melakukan inisialisasi maksimal iterasi dimana maksimal iterasi dapat berubah sesuai dengan jumlah iterasi yang ingin dilakukan. Pada kode program Gambar 5.15 dapat di lihat kode program penentuan parameter yang akan digunakan.

```
14 %% Tabu Search Parameters
15
16 - MaxIt=50; % Maximum Number of Iterations
17
18 - TL=round(0.5*nAction); % Tabu Length
19
```

Gambar 5.15 Inisialisasi Parameter

5.8.2 Inisialisasi Solusi

Tahap kedua pada implementasi tabu search adalah penentuan inisialisasi solusi. Dimana solusi yang akan dibuat sama dengan solusi pada proses algoritma genetika. Pada kode program Gambar 5.16 ditunjukkan bahwa inisialisasi solusi *profit* di buat pada variable “sol.Cost”, inisialisasi solusi untuk urutan posisi kapal yang akan melakukan pengiriman atau muatan *container* kapal i ke tujuan j pada perjalanan ke k disimpan pada variable “sol.Position”, dan solusi yang menunjukkan header yaitu kapal apa, tujuan mana dan perjalanan seberapa di simpan pada variable “sol.Head”. Setelah itu membuat inisialisasi solusi terbaik yang akan disimpan pada variable “BestSol”. Selain melakukan inisialisasi solusi, juga dilakukan inisialisasi *tabu counter* yang di inisialisasi pada variable “TC”.

```

21      %% Initialization
22
23      % Create Empty Individual Structure
24 -     empty_individual.Position=[];
25 -     empty_individual.Cost=-inf;
26 -     empty_individual.Head=[];
27
28      % Create Initial Solution
29 -     sol=empty_individual;
30 -     sol.Position=bestX1(4,:);
31 -     sol.Head=bestX1(1:3,:);
32 -     sol.Cost=CostFunction(sol.Head,sol.Position,Pij,ub,demand);
33
34      % Initialize Best Solution Ever Found
35 -     BestSol=empty_individual;
36
37      % Array to Hold Best Costs
38 -     BestCost=zeros(MaxIt,1);
39
40      % Initialize Action Tabu Counters
41 -     TC=zeros(nAction,1);

```

Gambar 5.16 Inisialisasi Solusi

5.8.3 Menghasilkan Solusi Terbaik

Setelah melakukan inisialisasi solusi, selanjutnya melakukan proses iterasi pada *tabu search*. Pada kode program Gambar 5.17 dijelaskan proses yang akan di lakukan untuk

menghasilkan solusi terbaik. Proses pencarian solusi dapat dihasilkan dengan memanggil fungsi “DoDistribute”, “DoAction” dan “CostFunction”. Pada fungsi “DoDistribute” digunakan untuk memecah muatan yang jumlahnya kurang dari setengah kapasitas kapal dan membagi rata muatan tersebut pada kapal-kapal yang memiliki tujuan yang sama. Fungsi “DoAction” digunakan untuk melakukan proses *swap*, *reversion* dan *insert* dimana muatan atau nilai variable setiap gen akan di lakukan proses *swap*, *reversion* dan *insert*. Sedangkan untuk fungsi “CostFunction” digunakan untuk mencari solusi *profit* terbaik berdasarkan pada kode program pada file “MyCost”. Proses pencarian solusi terus di ulang sampai mencapai nilai maksimal iterasi. Pada kode program baris 62, dijelaskan apabila solusi yang dihasilkan lebih baik dari solusi awal yang ada maka solusi tersebut merupakan solusi terbaik dan akan di simpan yang prosesnya akan di jelaskan pada tahap selanjutnya.

```

43 % Tabu Search Main Loop
44
45 for it=1:MaxIt
46
47     lowcap=find(and(sol.Position<(ub(1,:)./2),sol.Position>0));
48     fill_ship=find(sol.Position>(ub(1,:)./2));
49     bestnewsol.Cost=-inf;
50
51     % Apply Actions
52     for i=1:nAction
53         if TC(i)==0
54             Position1=DoDistribute(sol.Position,sol.Head,lowcap,fill_ship);
55             newsol.Position=DoAction(Position1,ActionList{i});
56             newsol.Head=sol.Head;
57             newsol.Cost=CostFunction(newsol.Head,newsol.Position,Fij,ub,demand);
58             newsol.ActionIndex=i;
59             lowcap=find(and(sol.Position<(ub(1,:)./2),sol.Position>0));
60             fill_ship=find(sol.Position>(ub(1,:)./2));
61
62             if newsol.Cost>=bestnewsol.Cost
63                 bestnewsol=newsol;
64             end
65         end
66     end

```

Gambar 5.17 Solusi Terbaik

5.8.4 Update Solusi dan Tabu List

Setelah memperoleh hasil atau solusi terbaik dari proses iterasi pada tahap sebelumnya, selanjutnya dilakukan *update* solusi berdasarkan solusi yang diperoleh dari proses iterasi dan

disimpan pada variable “sol”. Solusi tersebut masih dalam bentuk solusi untuk header, muatan kapal dan biaya. Kemudian dilakukan penambahan pada *tabu list* dan melakukan pengurangan pada *tabu counter*. Jika solusi yang dihasilkan memiliki *profit* lebih besar dari *profit* yang dimiliki pada solusi yang sudah ada maka solusi tersebut disimpan pada variable “BestSol” hal ini ditunjukkan pada kode program pada baris ke 80-82 pada Gambar 5.18. Selanjutnya melakukan penyimpanan solusi *profit* terbaik sesuai iterasi keberapa ke dalam variable “BestCost”

```

68      % Update Current Solution
69      sol=bestnewsol;
70
71      % Update Tabu List
72      for i=1:nAction
73          if i==bestnewsol.ActionIndex
74              TC(i)=TL;                % Add To Tabu List
75          else
76              TC(i)=max(TC(i)-1,0);    % Reduce Tabu Counter
77          end
78      end
79
80      % Update Best Solution Ever Found
81      if sol.Cost>=BestSol.Cost
82          BestSol=sol;
83      end
84
85      % Save Best Cost Ever Found
86      BestCost(it)=BestSol.Cost;

```

Gambar 5.18 Update Tabu list dan Solusi

Setelah melakukan tahap implementasi algoritma genetika dana *tabu search* pada kode program di atas, hasil *running* pada MATLAB untuk proses GA dapat di lihat pada lampiran E dan untuk proses GA-TS dapat dilihat pada lampiran F.

(Halaman ini sengaja dikosongkan)

BAB VI

HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan mengenai hasil dari proses uji coba dan pembahasan serta analisis terhadap hasil yang diperoleh dari proses implementasi kombinasi algoritma genetika dan pencarian tabu dalam penyelesaian permasalahan penyalokasian kapal dalam proses pengiriman barang.

6.1 Hasil Percobaan

Pada sub bab ini akan dijelaskan hasil percobaan yang diperoleh dari proses uji coba program MATLAB yang telah dibuat untuk menyelesaikan permasalahan alokasi kapal menggunakan kombinasi metode algoritma genetika dan *tabu search*.

6.1.1 Lingkungan Uji Coba

Lingkungan uji coba membahas mengenai lingkungan pengujian yang digunakan untuk implementasi tugas akhir ini meliputi perangkat keras dan perangkat lunak yang digunakan. Spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam implementasi algoritma genetika ditunjukkan pada Tabel 6.1

Tabel 6.1 Lingkunga Uji Coba

Perangkat Keras	Spesifikasi
Laptop	ASUS VivoBook X202E
Tipe Prosesor	Intel Core i3 Processor
Max. Memori	4 GB (1 DIMM)
Hard Drive Type	500 GB HDD
Perangkat Lunak	Fungsi
Windows 8.1	Sistem Operasi
Matlab R2013a	Melakukan Optimasi

Perangkat Lunak	Fungsi
Microsoft Excel 2013	Mengolah data Analisis Hasil Uji Coba

6.1.2 Parameter dan Sekenario Uji Coba

Pada proses uji coba akan digunakan parameter yang dapat berpengaruh pada proses implementasi algoritma genetika dan *tabu search*. Parameter tersebut adalah parameter probabilitas *crossover*, probabilitas mutasi dan jumlah populasi yang ingin dihasilkan. Perubahan input parameter akan mempengaruhi hasil atau solusi dari proses implementasi algoritma genetika dan *tabu search*. Penggunaan parameter sebagai inputan pada proses algoritma genetika dan *tabu search* dilakukan untuk mengetahui bagaimana kombinasi-kombinasi dari parameter yang dapat menghasilkan total keuntungan yang paling tinggi. Parameter-parameter yang digunakan sebagai perbandingan hasil algoritma genetika dan *tabu search* akan digunakan untuk mengathui beberapa hal berikut :

- Mengetahui perbandingan pengaruh banyaknya populasi (N) pada *profit* yang dihasilkan.
- Mengetahui perbandingan pengaruh nilai probabilitas pindah silang (Pc) pada *profit* yang dihasilkan.
- Mengetahui perbandingan pengaruh nilai probabilitas mutasi (Pm) pada *profit* yang dihasilkan.
- Mengetahui perbandingan pengaruh implementasi kombinasi metode *tabu search* pada *profit* yang dihasilkan.

Setelah menentukan parameter yang akan digunakan pada saat uji coba, selanjutnya perlu dibuat sekenario uji coba untuk mengetahui hasil algoritma genetika dan *tabu search* yang paling optimal. Uji coba yang dicatat dalam tugas ini menggunakan maksimum iterasi 300. Beberapa skenario tersebut antara lain sebagai berikut.

- a. Perbandingan *profit* yang diperoleh berdasarkan parameter jumlah populasi (N) sebanyak 150, 300 dan 450 pada setiap percobaan.
- b. Perbandingan total *profit* berdasarkan parameter Pc yaitu 0.8 dan 0.9 pada setiap percobaan. Penggunaan nilai parameter tersebut berdasarkan nilai probabilitas crossover yang disarankan yaitu berkisar antara 80 % - 95 % [17].
- c. Perbandingan *profit* yang diperoleh berdasarkan parameter Pm yaitu 0,05 dan 0,1 pada setiap percobaan. Penggunaan nilai parameter tersebut berdasarkan nilai probabilitas mutasi yang disarankan antara 0.5% - 1% [17].
- d. Perbandingan pengaruh penggunaan kombinasi metode *tabu seacrh* pada setiap percobaan.

6.1.3 Hasil Uji Coba

Pembahasan hasil uji coba akan dilakukan dengan melakukan perbandingan *profit* yang diperoleh dari seluruh percobaan. Uji coba dilakukan untuk mengetahui *profit* yang paling tinggi dengan mencoba parameter yang berbeda-beda. Selain itu uji coba dilakukan pada metode algoritma genetika (GA) dan kombinasi metode algoritma genetika dan *tabu seacrh* (GA-TS). Sehingga nantinya dapat dilakukan perbandingan antara kedua metode tersebut.

- a. Tabel 6.2 merupakan hasil percobaan menggunakan algoritma genetika dengan parameter N 150, 300, 450 , Pm 0.05 , dan Pc 0.8 diperoleh hasil terbaik ketika menggunakan parameter N 450, Pc 0,8 dan Pm 0,05.

Tabel 6.2 Uji Coba GA 1

Pc	Pm	N	Nilai Fitness GA (Rp)
0.8	0.05	150	12,881,400,000

Pc	Pm	N	Nilai Fitness GA (Rp)
0.8	0.05	300	12,290,600,000
0.8	0.05	450	13,092,133,333

- b. Tabel 6.3 hasil percobaan menggunakan algoritma genetika dengan parameter N 150, 300, 450 , Pm 0.1 , dan Pc 0.8. Diperoleh hasil terbaik ketika menggunakan parameter N 450, Pc 0,8 dan Pm 0,1

Tabel 6.3 Uji Coba GA 2

Pc	Pm	N	Nilai Fitness GA (Rp)
0.8	0.1	150	13,092,133,333
0.8	0.1	300	14,020,116,667
0.8	0.1	450	14,090,600,000

- c. Tabel 6.4 merupakan hasil percobaan menggunakan algoritma genetika dengan parameter N 150, 300, 450 , Pc 0.9 dan Pm 0.05. Diperoleh hasil terbaik ketika menggunakan parameter N 150, Pc 0,9 dan Pm 0,05

Tabel 6.4 Uji Coba GA 3

Pc	Pm	N	Nilai Fitness GA (Rp)
0.9	0.05	150	14,310,733,333
0.9	0.05	300	14,092,866,667
0.9	0.05	450	12,208,300,000

- d. Tabel 6.5 merupakan hasil percobaan menggunakan algoritma genetika dengan parameter N 150, 300, 450 , Pc

0.9 dan Pm 0.1 Diperoleh hasil terbaik ketika menggunakan parameter N 300, Pc 0,9 dan Pm 0,1

Tabel 6.5 Uji Coba GA 4

Pc	Pm	N	Nilai Fitness GA (Rp)
0.9	0.1	150	12,106,983,333
0.9	0.1	300	13,079,300,000
0.9	0.1	450	13,051,533,333

- e. Tabel 6.6 merupakan hasil percobaan menggunakan algoritma genetika dan *tabu search* dengan parameter N 150, 300, 450 , Pc 0.8, dan Pm 0.05. Diperoleh hasil terbaik ketika menggunakan parameter N 300, Pc 0,8 dan Pm 0,05

Tabel 6.6 Uji Coba GA-TS 1

Pc	Pm	N	Nilai Fitness GA-TS (Rp)
0.8	0.05	150	22,502,150,000
0.8	0.05	300	24,133,216,667
0.8	0.05	450	21,164,716,667

- f. Tabel 6.7 merupakan hasil percobaan menggunakan algoritma genetika dan *tabu search* dengan parameter N 150, 300, 450 , Pc 0.8, dan Pm 0.1. Diperoleh hasil terbaik ketika menggunakan parameter N 450, Pc 0,8 dan Pm 0,1

Tabel 6.7 Uji Coba GA-TS 2

Pc	Pm	N	Nilai Fitness GA-TS (Rp)
0.8	0.1	150	21,164,716,667

Pc	Pm	N	Nilai Fitness GA-TS (Rp)
0.8	0.1	300	23,092,666,667
0.8	0.1	450	24,574,866,667

- g. Tabel 6.8 merupakan hasil percobaan menggunakan algoritma genetika dan *tabu search* dengan parameter N 150, 300, 450 , Pc 0.9 dan Pm 0.05. Diperoleh hasil terbaik ketika menggunakan parameter N 450, Pc 0,9 dan Pm 0,05

Tabel 6.8 Uji Coba GA-TS 3

Pc	Pm	N	Nilai Fitness GA-TS (Rp)
0.9	0.05	150	24,410,900,000
0.9	0.05	300	21,468,283,333
0.9	0.05	450	24,202,633,333

- h. Tabel 6.9 merupakan hasil percobaan menggunakan algoritma genetika dan *tabu search* dengan parameter N 150, 300, 450 , Pc 0.9 dan Pm 0.1. Diperoleh hasil terbaik ketika menggunakan parameter N 150, Pc 0,9 dan Pm 0,1

Tabel 6.9 Uji Coba GA-TS 4

Pc	Pm	N	Nilai Fitness GA-TS (Rp)
0.9	0.1	150	23,293,050,000
0.9	0.1	300	21,769,066,667
0.9	0.1	450	22,452,550,000

6.1.4 Sekenario Uji Coba Terbaik

Berdasarkan hasil uji coba di atas, di dapatkan hasil terbaik dari masing-masing percobaan dengan menggunakan parameter yang berbeda-beda. Hasil terbaik dari masing-masing percobaan di tampilkan pada sekenario pada Tabel 6.10. Dapat dilihat pada tabael bahwa hasil terbaik dari percobaan menggunakan parameter yang berbeda-beda didapatkan parameter terbaik yang menghasilkan GA optimal adalah Pc 0,9 dan Pm 0,05 dengan jumlah populasi 150 sedangkan untuk menghasilkan GA-TS yang optimal adalah dengan menggunakan parameter Pc 0,8 dan Pm 0,1 dengan jumlah populasi 450. Sehingga kombinasi parameter tersebut akan di gunakan kembali untuk dilakukan uji coba sebanyak sepeuluh kali *running* pada masing-masing metode GA dan GA-TS yang kemudian hasilnya akan di analisis.

Tabel 6.10 Sekenario Terbaik

Pc	Pm	N	Metode	Nilai Fitness (Rp)
0.8	0.05	450	GA	13,092,133,333
0.8	0.1	450	GA	14,090,600,000
0.9	0.05	150	GA	14,310,733,333
0.9	0.1	300	GA	13,079,300,000
0.8	0.05	300	GA-TS	24,133,216,667
0.8	0.1	450	GA-TS	24,574,866,667
0.9	0.05	450	GA-TS	24,202,633,333
0.9	0.1	150	GA-TS	23,293,050,000

6.1.5 Proses *Running* GA dan GA-TS

Hasil kombinasi terbaik dari beberapa parameter yang digunakan akan di lakukan *running* sebanyak sepuluh kali pada MATLAB dan hasil *random* muatan yang diperoleh akan

dihitung menggunakan Microsoft Excel 2013 untuk mengetahui solusi terbaik yang dapat digunakan untuk penyelesaian permasalahan pada tugas akhir ini. Pada bagian ini akan dilakukan perhitungan total aktual *profit*, total *opportunity cost*, total *net profit*, dan waktu komputasi (detik). Berdasarkan hasil dari sepuluh proses *running* tersebut akan diambil solusi terbaik untuk menjadi solusi penyelesaian permasalahan. Berikut adalah solusi terbaik yang dipilih dari hasil *running* sebanyak sepuluh kali.

a. Hasil *running* GA

Tabel 6.11 adalah hasil percobaan dengan melakukan *running* sebanyak sepuluh kali menggunakan kombinasi parameter terbaik yang dihasilkan dari sekenario uji coba GA yaitu dengan jumlah populasi 150, Pc 0,9, dan Pm 0,05. Solusi terbaik yang dihasilkan terdapat pada uji coba dari proses *running* keenam.

Tabel 6.11 Running GA

Running Ke	Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)	Waktu Komputasi (detik)
1	23,746,583,333	10,997,016,667	12,749,566,667	111.271
2	20,319,483,333	8,751,433,333	11,568,050,000	110.461
3	24,680,100,000	10,950,600,000	13,729,500,000	110.265
4	22,447,750,000	10,648,316,667	11,799,433,333	110.172
5	25,430,433,333	12,814,900,000	12,615,533,333	115.573
6	25,620,100,000	10,936,100,000	14,684,000,000	115.323

Running Ke	Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)	Waktu Komputasi (detik)
7	24,887,733,333	12,045,083,333	12,842,650,000	111.672
8	21,570,666,667	7,703,733,333	13,866,933,333	112.221
9	20,692,616,667	7,768,216,667	12,924,400,000	111.532
10	21,598,150,000	7,804,500,000	13,793,650,000	113.595

b. Hasil *running* GA-TS

Tabel 6.12 adalah hasil percobaan dengan melakukan *running* sebanyak sepuluh kali menggunakan kombinasi parameter terbaik yang dihasilkan dari sekenario uji coba GA-TS yaitu dengan jumlah populasi 450, Pc 0,8, dan Pm 0,1. Solusi terbaik yang dihasilkan terdapat pada uji coba dari proses *running* kesembilan dapat dilihat pada tabel

Tabel 6.12 Running GA-TS

Running Ke	Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)	Waktu Komputasi (detik)
1	24,266,333,333	3,232,833,333	21,033,500,000	506,631
2	25,320,783,333	3,924,533,333	21,396,250,000	652,375
3	30,019,766,667	7,283,866,667	22,735,900,000	533.436
4	25,282,833,333	3,920,100,000	21,362,733,333	608.208

Running Ke	Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)	Waktu Komputasi (detik)
5	27,105,566,667	5,637,283,333	21,468,283,333	556.867
6	26,679,533,333	2,821,133,333	23,858,400,000	465.579
7	27,671,000,000	4,178,200,000	23,492,800,000	551.356
8	24,908,116,667	3,743,400,000	21,164,716,667	655.056
9	29,828,833,333	5,080,233,333	24,748,600,000	665.752
10	27,007,433,333	4,399,166,667	22,608,266,667	516.529

6.1.6 Validasi

Bagian validasi dilakukan untuk memastikan hasil komputasi sesuai dengan tujuan yang diharapkan. Validasi ini dilakukan dengan membandingkan data hasil keluaran model pada MATLAB dengan perhitungan matematis yang dilakukan pada data perusahaan. Perbandingan tersebut harus menunjukkan bahwa data hasil optimasi sesuai atau berada pada tingkat kesalahan (error) yang diperbolehkan sehingga model optimasi dapat dikatakan valid. Untuk mengetahui error yang dimiliki, pertama dilakukan perhitungan *standard deviasi* terlebih dahulu pada data perusahaan dan pada model hasil keluaran MATLAB. Perhitungan dilakukan pada *profit* masing-masing yang diperoleh kapal dengan menggunakan tarif yang dibuat sama sebagai pembanding. Tabel 6.13 dapat dilihat perhitungan matematis pada data perusahaan. Setelah dilakukan perhitungan *standard deviasi* diperoleh hasil sebesar 513248387.8 pada data perusahaan.

Tabel 6.13 Profit Data Perusahaan

Tujuan	Kapal	Kapasitas	Muatan	Tarif (Rp)	Nilai Fitness (Rp)
1	2	296	80	4,800,000	652,800,000
1	1	650	350	1,800,000	90,000,000
1	1	650	350	1,800,000	90,000,000
1	1	650	350	1,800,000	90,000,000
1	1	650	350	1,800,000	90,000,000
1	1	650	350	1,800,000	90,000,000
1	1	650	350	1,800,000	90,000,000
2	6	406.00	130.00	5,033,333	734,866,666
2	7	538	250.00	5,700,000.00	216,600,000
2	8	537	150.00	5,700,000.00	1,350,900,000
2	8	538	250.00	5,700,000.00	216,600,000
2	7	538	250.00	5,700,000.00	216,600,000
2	8	538	250.00	5,700,000.00	216,600,000
3	9	330	150	6,200,000.00	186,000,000
3	8	518	150	5,700,000.00	1,242,600,000
3	9	330	150	6,200,000.00	186,000,000
3	8	518	150	5,700,000.00	1,242,600,000
3	9	330	150	6,200,000.00	186,000,000
3	8	518	150	5,700,000.00	1,242,600,000
4	12	219	140	2,750,000.00	167,750,000
4	12	219	140	2,750,000.00	167,750,000
4	12	219	140	2,750,000.00	167,750,000
4	12	219	140	2,750,000.00	167,750,000
4	12	219	140	2,750,000.00	167,750,000

Tujuan	Kapal	Kapasitas	Muatan	Tarif (Rp)	Nilai Fitness (Rp)
4	12	219	140	2,750,000.00	167,750,000

Setelah melakukan perhitungan matematis pada data perusahaan dihitung pula secara manual *profit* dari masing-masing kapal pada model keluaran MATLAB. Pada tabel 6.14 perhitungan *profit* masing –masing kapal. Setelah dilakukan perhitungan *standard deviasi* diperoleh hasil 554573195.2 pada perhitungan model keluaran MATLAB.

Tabel 6.14 Validasi GA-TS

Tujuan	Kapal	Kapasitas	Muatan	Tarif (Rp)	Nilai Fitness (Rp)
1	6	406	356	5,033,333	1,540,200,000
1	6	406	359	5,033,333	1,570,400,000
1	8	538	417	5,700,000	1,687,200,000
1	8	538	466	5,700,000	2,245,800,000
1	9	330	322	6,200,000	1,946,800,000
1	10	283	260	6,200,000	1,469,400,000
2	8	538	531	5,700,000	2,986,800,000
2	9	330	297	6,200,000	1,636,800,000
2	10	283	175	6,200,000.00	415,400,000
2	10	283	277	6,200,000.00	1,680,200,000
3	8	538	440	5,700,000.00	1,949,400,000
3	8	538	460	5,700,000.00	2,177,400,000
4	8	538	415	5,700,000.00	1,664,400,000
4	8	538	425	5,700,000.00	1,778,400,000

Setelah mendapat hasil perhitungan matematis yang telah diperlihatkan pada tabel 6.13 dan tabel 6.14 dan *standard deviasi* yang dihasilkan dari masing-masing perhitungan yaitu untuk *standard deviasi* pada data perusahaan sebesar

513248387.8 dan pada model GA-TS adalah 554573195.2. Selanjutnya dilakukan perbandingan dan menghitung nilai *error variance*. Berdasarkan perhitungan menggunakan rumus *error variance* diperoleh error sebesar 0.080516195 atau 8% dapat dilihat pada perhitungan berikut

$$E = \frac{|Standard\ deviasi\ simulasi - Standard\ deviasi\ data|}{Standard\ deviasi\ data}$$

$$E = \frac{|554573195.2. - 513248387.8|}{513248387.8}$$

$$E = 0.080516195$$

$$E = 8\%$$

6.2 Pembahasan dan Analisis

Pada sub bab ini akan dijelaskan pembahasan dan analisis dari hasil semua proses uji coba yang telah di lakukan dengan penggunaan parameter yang berbeda-beda. Selain di lakukan analisis juga akan dilakukan perbandingan hasil metode GA dan GA-TS.

6.2.1 Analisis Hasil Uji Coba Algoritma Genetika

Berdasarkan hasil uji coba implementasi menggunakan algoritma genetika, dapat di lihat pada Tabel 6.10 sekenario hasil terbaik dari percobaan algoritma genetika ditunjukkan pada penggunaan parameter jumlah populasi 150 , Pc 0.9 dan Pm 0.05 dengan total *profit* terbaik atau nilai *fitness* terbaik yaitu Rp14,310,733,333. Sedangkan berdasarkan hasil *running* GA sebanyak sepuluh kali menghasilkan solusi terbaik dengan *profit* atau *fitness* yaitu Rp 14,684,000,000.00. Semua kromosom yang diperoleh dari asil *running* sebanyak sepuluh kali dapat di lihat pada lampiran A.

Pada Tabel 6.15 merupakan potongan kromosom hasil *running* GA terbaik yang menunjukkan kapal i menuju tujuan j pada perjalanan ke k dengan jumlah muatan yang dibawa.

Berdasarkan hasil *running* didapat solusi terbaik oleh metode algoritma genetika yaitu untuk melakukan pengiriman *container* menuju tujuan 1 dengan jumlah permintaan sebanyak 2180 *container* dapat dilakukan dengan mengalokasikan kapal 1 membawa muatan 149 *container*, kapal 7 membawa 23 *container*, kapal 8 perjalanan ke satu membawa 473 *container*, kapal 8 perjalanan ke dua membawa 535 *container*, kapal 10 perjalanan ke satu membawa 262 *container*, kapal 10 perjalanan kedua membawa 269 *container*, kapal 10 perjalanan ketiga membawa 283 *container* dan kapal 11 membawa muatan 186 .

Tabel 6.15 GA Muatan Tujuan 1

Kapal	1	7	8	8	10	10	10	11
Tujuan	1	1	1	1	1	1	1	1
Slot	a	a	a	b	a	b	c	a
Kapasitas	650	538	538	538	283	283	283	211
Tarif	180 000 0	570 000 0	570 000 0	570 000 0	620 000 0	620 000 0	620 000 0	200 000 0
Muatan	149	23	473	535	262	269	283	186

Pada bagian Tabel 6.16 merupakan potongan kromosom kedua hasil *running* GA yang berisi solusi kapal yang akan digunakan untuk melakukan pengiriman *container* menuju tujuan 2 dengan jumlah permintaan pada tujuan 2 yaitu sebanyak 1280 *container*. Permintaan tersebut akan dipenuhi oleh kapal i yang membawa muatan j pada tujuan 2 yang dapat dilihat pada tabel berikut

Tabel 6.16 GA Muatan Tujuan 2

Kapal	1	2	7
Tujuan	2	2	2
Slot	a	a	a
Kapasitas	650	296	538
Tarif	1800000	4800000	5700000
Muatan	611	250	419

Pada Tabel 6.17 merupakan solusi kapal hasil *running* GA yang akan digunakan untuk melakukan pengiriman *container* menuju tujuan 3 dengan jumlah permintaan sebanyak 900 *container*, dan masing-masing jumlah muatan yang harus di bawa oleh masing-masing kapal i ke tujuan j dapat dilihat pada tabel berikut

Tabel 6.17 GA Muatan Tujuan 3

Kapal	7	8	10
Tujuan	3	3	3
Slot	a	a	a
Kapasitas	538	538	283
Tarif	5700000	5700000	6200000
Muatan	455	351	94

Pada Tabel 6.18 merupakan solusi kapal hasil *running* GA yang akan digunakan untuk melakukan pengiriman *container* menuju tujuan 4 dengan jumlah permintaan sebanyak 840 *container*, dan masing-masing jumlah muatan yang harus di bawa oleh masing-masing kapal i ke tujuan j dapat dilihat pada tabel berikut

Tabel 6.18 GA Muatan Tujuan 4

Kapal	2	8	9	11
Tujuan	4	4	4	4
Slot	a	a	a	a
Kapasitas	296	538	330	211
Tarif	4800000	5700000	6200000	2000000
	206	205	280	149

Pada Tabel 6.19 merupakan hasil perhitungan Total Aktual Profit, Total Opportunity Cost, Total Net Profit dan Waktu Komputasi (detik) dari semua kapal yang menuju tujuan Makassar, Bitung, Ambon dan Banjarmasin. Total *opportunity coost* yang bernilai cukup besar menyebabkan keuntungan atau *profit* yang diperoleh perusahaan mengalami pengurangan yang cukup banyak. Perhitungan tersebut merupakan hasil terbaik yang daiperoleh dari hasil *running* GA .

Tabel 6.19 GA Profit

Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)	Waktu Komputasi (detik)
25,620,100,000	10,936,100,000	14,684,000,000	115.323 s

Berdasarkan pada Tabel 6.20 dijelaskan minimal, maksimal dan rata-rata dari *load factor* yang dibawa oleh semua kapal menuju tujuan Makassar, Bitung, Ambon dan Banjarmasin. Masih terdapat angka 4% yang menunjukkan bahwa muatan yang dibawa oleh kapal tertentu belum optimal. Proses ini di lakukan dengan metode algortima genetika, diharapkan denga melakukan kombinasi metode algoritma genetika dan *tabu*

search pada tahapan selanjutnya dapat memaksimalkan *load factor* dan memuat *container* lebih optimal.

Tabel 6.20 Load Factor hasil GA

Min Load Factor	Max Load Factor	Rata-Rata Load Factor
4%	100%	72%

6.2.2 Analisis Hasil Uji Coba Kombinasi Algoritma Genetika dan *Tabu Search*

Berdasarkan hasil uji coba implementasi menggunakan algoritma genetika dan *tabu search*, dapat di lihat pada Tabel 6.10 sekenario hasil terbaik dari percobaan algoritma genetika dan *tabu seacrh* ditunjukkan pada penggunaan parameter jumlah populasi 450 , Pc 0.8 dan Pm 0.1 dengan total *profit* terbaik yaitu Rp 24,574,866,667. Sedangkan berdasarkan hasil running GA-TS sebanyak sepuluh kali menghasilkan solusi terbaik dengan *profit* yaitu Rp 24,748,600,000.00. Semua kromosom yang diperoleh dari asil *running* sebanyak sepuluh kali dapat di lihat pada lampiran B.

Pada Tabel 6.21 merupakan potongan kromosom hasil *running* GA-TS yang menunjukkan kapal i menuju tujuan j pada perjalanan ke k dengan jumlah muatan yang dibawa. Berdasarkan hasil *running* didapat solusi terbaik oleh metode kombinasi algortima genetika dan *tabu search* yaitu untuk melakukan pengiriman *container* menuju tujuan 1 dapat di alokasikan kapal 6 perjalanan kesatu dengan muatan 356 *container*, kapal 6 perjalanan kedua dengan muatan 358 *container*, kapal 8 perjalanan ke satu dengan muatan 417 *container*, kapal 8 perjalanan kedua dengan muatan 466 *container*, kapal 9 dengan muatan 322 *container* dan kapal 10 dengan muatan 260 *container*. Hasil tersebut merupakan hasil uji coba dengan melakukan *running* sebanyak sepuluh kali dan solusi yang di ambil merupakan hasil dari proses *running* keenam.

Tabel 6.21 GA-TS Tujuan 1

Kapal	6	6	8	8	9	10
Tujuan	1	1	1	1	1	1
Slot	a	b	a	b	a	a
Kapasitas	406	406	538	538	330	283
Tarif	50333 33	50333 33	57000 00	57000 00	62000 00	62000 00
Muatan	356	359	417	466	322	260

Pada bagian Tabel 6.22 merupakan potongan kromosom kedua hasil *running* GA-TS yang berisi solusi kapal yang akan digunakan untuk melakukan pengiriman container menuju tujuan 2 dengan jumlah permintaan pada tujuan 2 yaitu sebanyak 1280 container. Permintaan tersebut akan dipenuhi oleh kapal i yang membawa muatan j pada tujuan 2 yang dapat dilihat pada tabel berikut

Tabel 6.22 GAT-TS Tujuan 2

Kapal	8	9	10	10
Tujuan	2	2	2	2
Slot	a	a	a	b
Kapasitas	538	330	283	283
Tarif	5700000	6200000	6200000	6200000
Muatan	531	297	175	277

Pada Tabel 6.23 merupakan solusi kapal hasil *running* GA-TS yang akan digunakan untuk melakukan pengiriman *container* menuju tujuan 3 dengan masing-masing jumlah muatan yang dapat dilihat pada tabel berikut

Tabel 6.23 GA-TS Tujuan 3

Kapal	8	8
Tujuan	3	3
Slot	a	b
Kapasitas	538	538
Tarif	5700000	5700000
9	440	460

Pada Tabel 6.24 merupakan solusi kapal hasil *running* GA-TS yang akan digunakan untuk melakukan pengiriman *container* menuju tujuan 4 dengan masing-masing jumlah muatan yang dapat dilihat pada tabel berikut

Tabel 6.24 GA-TS Tujuan 4

Kapal	8	8
Tujuan	4	4
Slot	a	b
Kapasitas	538	538
Tarif	5700000	5700000
9	415	425

Pada Tabel 6.25 merupakan hasil perhitungan Total Aktual Profit, Total Opportunity Cost, Total Net Profit dan Waktu Komputasi (detik) dari semua kapal yang menuju tujuan Makassar, Bitung, Ambon dan Banjarmasin. Total opportunity cost yang bernilai cukup kecil menyebabkan keuntungan atau *profit* yang diperoleh perusahaan tidak mengalami pengurangan yang banyak. Perhitungan tersebut merupakan hasil terbaik yang diperoleh dari hasil *running* GA-TS.

Tabel 6.25 GA-TS Profit

Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)	Waktu Komputasi (detik)
29,828,833,333	5,080,233,333	24,748,600,000	666

Berdasarkan pada Tabel 6.26 dijelaskan minimal, maksimal dan rata-rata dari *load factor* yang dibawa oleh semua kapal. Dapat dilihat pada tabel bahwa minimal *load factor* yang dibawa adalah 62%, ini menunjukkan perbaikan jika dibandingkan dengan menggunakan metode algoritma genetika saja yang menunjukkan minimal *load factor* masih 4%. Penggunaan kombinasi algoritma genetika dan *tabu search* menunjukkan peningkatan dan dapat membawa muatan lebih optimal dibandingkan menggunakan algoritma genetika saja.

Tabel 6.26 Load Factor hasil GA-TS

Min Load Factor	Max Load Factor	Rata-Rata Load Factor
62%	99%	86%

6.2.3 Perbandingan Hasil Algoritma Genetika dengan Algoritma Genetika dan *Pencarian Tabu*

Setelah melakukan analisis dari kedua implementasi metode GA dan GA-TS dengan melakukan uji coba parameter terbaik dan melakukan uji coba lagi dengan melakukan *running* sebanyak sepuluh kali menggunakan parameter yang telah ditetapkan sebagai parameter terbaik. Selanjutnya dapat dilakukan perbandingan dari hasil kedua algoritma tersebut untuk mengetahui keuntungan atau *profit* terbaik yang didapatkan yang nantinya akan menjadi pertimbangan pihak perusahaan pelayaran dalam melakukan perencanaan pengalokasian kapal yang akan digunakan untuk mengirim barang atau *container*. Pada tugas akhir ini memiliki tujuan untuk mengalokasikan kapal dengan memaksimalkan muatan

dan keuntungan atau *profit* perusahaan. Sehingga dua hal tersebut akan menjadi pembandingan untuk menentukan metode yang terbaik dalam mencapai tujuan tersebut. Pada Tabel 6.27 dapat dilihat bahwa *profit* yang diperoleh menggunakan metode GA-TS lebih besar dari pada menggunakan GA saja. Setelah dilakukan perhitungan metode GA-TS memiliki *profit* 69% lebih besar dibanding pada saat menggunakan metode GA. Selain disebabkan karena nilai random yang di hasilkan pada masing-masing kromosom lebih baik pada proses penerapan GA-TS. Hal ini juga disebabkan biaya *opportunity cost* pada proses GA lebih besar dibandingkan pada proses GA-TS.

Tabel 6.27 Perbandingan Profit GA & GA-TS

Metode	Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)
GA	25,620,100,000	10,936,100,000	14,684,000,000
GA-TS	29,828,833,333	5,080,233,333	24,748,600,000

Setelah melakukan perbandingan *profit* yang diperoleh pada masing-masing metode, selanjutnya membandingkan muatan yang lebih optimal pada masing-masing metode GA dan GA-TS. Pada Tabel 6.28 dapat dilihat perbandingan *load factor* pada masing-masing metode. *Load factor* merupakan hasil pembagian antara muatan yang dibawa dibagi dengan kapasitas maksimum kapal. Persentase *load factor* hasil GA-TS lebih besar 14% dibandingkan dengan menggunakan metode GA saja. Hal ini disebabkan karena pada metode GA terdapat nilai minimal *load factor* yang terlalu kecil yang menunjukkan bahwa muatan yang dibawa kapal tertentu masih belum optimal. Sedangkan dengan menggunakan GA-TS muatan sudah lebih optimal dengan minimal *load factor* tidak kurang dari 50%

Tabel 6.28 Perbandingan Load Factor GA dan GA-TS

Metode	Min Load Factor	Max Load Factor	Rata-Rata Load Factor
GA	4%	100%	72%
GA-TS	62%	99%	86%

6.2.4 Perbandingan sebelum dan sesudah implementasi GA-TS

Pada tahap perbandingan sebelum dan sesudah implementasi GA-TS dilakukan perhitungan pada data yang diperoleh dari perusahaan dan hasil terbaik dari GA-TS berdasarkan muatan yang dibawa oleh masing-masing kapal. Setelah dilakukan perhitungan untuk total actual cost, total *opportunity cost*, dan total *profit* yang diperoleh perusahaan sebelum dan sesudah implementasi GA-TS dapat dilihat pada Tabel 6.29 bahwa total *profit* yang diperoleh perusahaan lebih tinggi dan optimal ketika dilakukan implementasi GA-TS. Metode GA-TS memiliki *profit* dengan peningkatan lebih dari 100% dibanding pada kondisi perusahaan yaitu sebelum menggunakan GA-TS. Hal ini disebabkan karena pada data perusahaan, muatan yang dibawa oleh kapal-kapal yang digunakan untuk mengirimkan barang atau *container* masih belum optimal. Dengan kata lain masih banyak kapal yang membawa muatan tidak sampai setengah dari kapasitas kapal, sehingga menyebabkan nilai *opportunity cost* menjadi sangat besar dan menyebabkan nilai *profit* menjadi semakin kecil.

Permasalahan tersebut diselesaikan dalam tugas akhir ini menggunakan metode kombinasi algoritma genetika dan *tabu search*. Dimana muatan kapal i yang akan dibawa menuju tujuan j pada perjalanan ke k akan membawa muatan yang lebih optimal, sehingga *opportunity cost* tidak lagi bernilai besar dan dapat menghasilkan *profit* yang lebih maksimal. Hal ini juga ditunjukkan dengan adanya rata-rata *load factor* yang dihasilkan dari perhitungan data perusahaan yaitu 48%

dan berdasarkan perhitungan dari hasil GA-TS rata-rata *load factor* meningkat menjadi 86%. Hal ini menunjukkan bahwa GA-TS dapat memaksimalkan muatan 38% lebih baik dibanding sebelum menggunakan GA-TS. Sehingga hal ini dapat membantu perusahaan dalam membuat perencanaan pengalokasian kapal. Tabel 6.29 dapat dilihat perbandingan *profit* sebelum dan sesudah optimasi dan perbandingan *load factor* sebelum dan sesudah optimasi.

Tabel 6.29 Perbandingan Sebelum dan Sesudah Optimasi

Perbandingan	Total Aktual Profit (Rp)	Total Opportunity Cost (Rp)	Total Net Profit (Rp)
Sebelum penerapan metode GA-TS	25,382,600,000	19,038,333,333	6,344,266,666
Sesudah penerapan metode GA-TS	29,828,833,333	5,080,233,333	24,748,600,000
Perbandingan	<i>Min Load Factor</i>	<i>Max Load Factor</i>	<i>Rata-Rata Load Factor</i>
Sebelum penerapan metode GA-TS	27%	64%	48%
Sesudah penerapan metode GA-TS	62%	99%	86%

(Halaman ini sengaja dikosngkan)

BAB VII

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan yang didapat dari seluruh proses pengerjaan tugas akhir dan juga saran perbaikan untuk penelitian kedepannya untuk dapat dikembangkan dari tugas akhir ini.

7.1 Kesimpulan

Adapun beberapa hal yang dapat disimpulkan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Optimasi alokasi kapal dilakukan untuk memperoleh keuntungan yang optimal bagi pihak pengirim atau pihak perusahaan pelayaran dengan memaksimalkan muatan yang dibawa. Penerapan metode algoritma genetika dan *tabu search* memberikan hasil yang optimal dengan total keuntungan atau *profit* yang diperoleh perusahaan sebelum implementasi adalah Rp 6,344,266,666.67 dan sesudah implementasi GA-TS adalah Rp 24,748,600,000.00. Selain itu *load factor* yang dihasilkan dari perhitungan data perusahaan yaitu 48% dan berdasarkan penerapan GA-TS *load factor* meningkat menjadi 86%. Sehingga dapat disimpulkan bahwa metode GA-TS memberikan peningkatan dan dapat mengoptimalkan *profit* serta muatan yang harus dibawa dalam proses pengiriman barang atau *container*.
2. Pemilihan jumlah populasi, nilai probabilitas pindah silang dan nilai probabilitas mutasi memiliki peran penting dalam melakukan optimasi menggunakan metode algoritma genetika. Jumlah populasi (N) yang dapat memberikan hasil paling optimal pada permasalahan optimasi alokasi kapal adalah 450 di

antara jumlah populasi yang digunakan yaitu 150, 300 dan 450. Sedangkan nilai probabilitas pindah silang (P_c) dan nilai probabilitas mutasi (P_m) yang dapat memberikan hasil paling optimal pada permasalahan optimasi alokasi kapal adalah 0.8 dan 0.1.

3. Mengkombinasikan algoritma genetika dengan *tabu search* memberikan hasil yang paling optimal dibandingkan menggunakan algoritma genetika saja. Hal ini ditunjukkan dengan adanya peningkatan *profit* sebesar 69% dan peningkatan *load factor* sebesar 14% pada saat menggunakan GA-TS dibandingkan ketika menggunakan GA saja.
4. Waktu komputasi yang di butuhkan pada saat melakukan *running* GA-TS lebih lama yaitu 666 s dibandingkan waktu yang dibutuhkan untuk melakukan *running* GA yaitu 115.323 s. Hal ini disebabkan karena pada setiap iterasi solusi yang akan dievaluasi akan dicocokkan terlebih dahulu dengan isi tabu list, dimana tabu list memiliki jumlah yang cukup banyak sesuai dengan jumlah pergerakan dari solusi yang ada.

7.2 Saran

1. Beberapa variabel yang belum dipertimbangkan dalam penelitian ini dapat digunakan pada penelitian selanjutnya untuk dapat lebih menyempurnakan dan memeberi solusi yang lebih baik, misalnya seperti variable waktu perjalanan yang dibutuhkan, penggunaan bahan bakar dan jarak.
2. Jumlah variable keputusan yang terlalu banyak menyebabkan penggunaan metode algoritma genetika saja tidak dapat memebrica hasil yang optimal, sehingga

dapat di lakukan penggunaan kombinasi metode algoritma metaheuristik lainnya untuk mendapatkan hasil lebih optimal selain algoritma *tabu search*.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] G. Garai and B. B. Chaudhuri, "A novel hybrid genetic algorithm with Tabu search for optimizing multi-dimensional functions and point pattern recognition," *Inf. Sci.*, vol. 221, pp. 28–48, Feb. 2013 [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0020025512006056>. [Accessed: 05-Oct-2016]
- [2] D.-Y. Lin and H.-Y. Liu, "Combined ship allocation, routing and freight assignment in tramp shipping," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 47, no. 4, pp. 414–431, Jul. 2011 [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1366554510001237>. [Accessed: 05-Oct-2016]
- [3] K. Setemen, "Implementasi Algoritma Genetika dalam Pengembangan Sistem Aplikasi Penjadwalan Kuliah," *J. IKA*, vol. 8, no. 1, 2010 [Online]. Available: <http://ejournal.undiksha.ac.id/index.php/IKA/article/view/156>. [Accessed: 05-Oct-2016]
- [4] E. Suhartono, "OPTIMASI PENJADWALAN MATA KULIAH DENGAN ALGORITMA GENETIKA (Studi Kasus di AMIK JTC Semarang)," *INFOKAM*, vol. 11, no. 5, 2015 [Online]. Available: <http://jurnal.amikjtc.com/index.php/jurnal/article/view/86>. [Accessed: 30-Sep-2016]
- [5] Marwana, "Optimasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Berbasis Permintaan MahasiswaO," *Pros. Konf. Nas. Ilmu Komput. 2012*, 2012.
- [6] A. O. Adewumi and M. M. Ali, "A multi-level genetic algorithm for a multi-stage space allocation problem," *Math. Comput. Model.*, vol. 51, no. 1–2, pp. 109–126, Jan. 2010 [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0895717709003306>. [Accessed: 30-Sep-2016]

- [7] S. Anwar, "Mengenal jasa transportasi laut dan udara," *J. Transp.* 112 2010 1-9, 2010.
- [8] D. Ronen, "Cargo ships routing and scheduling: Survey of models and problems," *Eur. J. Oper. Res.*, vol. 12, no. 2, pp. 119–126, 1983 [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0377221783902151>. [Accessed: 01-Oct-2016]
- [9] F. Valentito, "Optimasi Skenario Bunkering dan Kecepatan Kapal pada Pelayaran Trampoer," *J. Tek. POMITS Vol 1 No 1 2012 1-5*.
- [10] X. Heli, Y. Shanshan, and Lijia, "The Application of Genetic Algorithm Based on Multi-dimension Code Scheme on Course Scheduling In Adult Education," *Proc. Third Int. Symp. Electron. Commer. Secur. Work. '10 Guangzhou P R China 29-31 July 2010 Pp 005-009*, 2010.
- [11] D. Hermawanto, "Algoritma Genetika dan contoh aplikasinya," *Retrieved*, vol. 10, no. 25, p. 2013, 2003 [Online]. Available: http://www.academia.edu/download/33039788/algoritma_genetika_dan_contoh_aplikasinya_-_denny_hermawanto.pdf. [Accessed: 30-Sep-2016]
- [12] M. Hanita, "Penerapan Algoritma Genetika pada Penjadwalan Mata Kuliah (Studi Kasus: Program Studi Matematika FMIPA Universitas Bengkulu)," *Non Publ. Univ. Bengkulu*, 2011.
- [13] L. Spector, "Autoconstructive evolution: Push, pushGP, and pushpop," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, 2001, vol. 137 [Online]. Available: <http://luisamark.com/mark/ou/docs/ace.pdf>. [Accessed: 01-Oct-2016]
- [14] H. M. Sitorus, C. P. Sitorus, and P. Ciputra, "Penerapan Algoritma Tabu Search Pada Permasalahan Lintasan Keseimbangan Bentuk U Tipe I Dengan Waktu Proses Stokastik," *Ind. Syst. Eng. Assess. J. INASEA*, vol. 15, no. 1, 2014 [Online]. Available:

- <http://journal.binus.ac.id/index.php/inasea/article/view/238>. [Accessed: 02-Jan-2017]
- [15] “Rumus Statistik: Juli 2013.” [Online]. Available: <https://www.rumusstatistik.com/2013/07>. [Accessed: 08-Jan-2017]
- [16] A. Mayasar, *Optimasi Alokasi Dermaga pada PT Pelabuhan Indonesia III (Persero) Cabang Tanjung Perak dengan Algoritma Genetika*. 2016.
- [17] W. Hartono, “OPTIMASI ALOKASI SUMBER DAYA DENGAN GENETIC ALGORITHMS,” *Media Tek. Sipil*, vol. 11, no. 2, pp. 108–114, 2016 [Online]. Available: <http://media.sipil.ft.uns.ac.id/index.php/mts/article/view/132>. [Accessed: 30-Sep-2016]
- [18] B. Santosa and P. Willy, *Metoda Metaheuristik Konsep dan implementasi*. Prima printing, Surabaya, 2011.

(Halaman ini sengaja dikosongkan)



BIODATA PENULIS

Penulis bernama lengkap Novian Tiandini, Penulis dilahirkan di Bangkalan, 27 November 1994. Penulis telah menempuh beberapa pendidikan formal, yaitu TK Seruni Bangkalan, SD Banyuajuh 03 Bangkalan pada tingkat sekolah dasar, SMP Negeri 2 Probolinggo pada tingkat sekolah menengah pertama, dan SMA Negeri 1 Probolinggo pada tingkat sekolah menengah atas. Setelah menerima kelulusan SMA pada tahun 2013, penulis mengikuti pendaftaran mahasiswa baru ITS, yang akhirnya terdaftar sebagai mahasiswi di jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya dari jalur SNMPTN dengan NRP 5213100075. Selama menjadi mahasiswa, penulis telah mengikuti berbagai kegiatan kemahasiswaan dan aktif sebagai staff Kesejahteraan Mahasiswa Himpunan Mahasiswa Sistem Informasi (HMSI) pada tahun kepengurusan 2014/2015 dan menjabat sebagai sekretaris umum Himpunan Mahasiswa Sistem Informasi (HMSI) pada tahun kepengurusan 2015/2016. Selain itu penulis juga pernah mengikuti Program Kreativitas Mahasiswa dibidang kewirausahaan yang diselenggarakan oleh Dikti dan termasuk salah satu PKM didanai tahun 2015.

Pada Jurusan Sistem Informasi, penulis mengambil bidang minat Laboratorium Rekayasa Data dan Inteligensi Bisnis (RDIB) dengan topik Optimasi, yakni optimasi pengalokasian kapal peti kemas. Penulis dapat dihubungi melalui email novian.andini075@gmail.com

(Halaman ini sengaja dikosongkan)

LAMPIRAN A

Lampiran ini adalah hasil analisis kromosom yang berisi muatan dari masing-masing kapal yang harus dibawa menuju tujuan satu yaitu Makassar, tujuan dua Bitung, tujuan tiga Ambon dan tujuan empat Banjarmasin. Kromosom-kromosom ini di hasilkan dari proses uji coba Algoritma Genetika dengan menggunakan parameter N 150, Pc 0.9, dan Pm 0,05 yang dilakukan sebanyak sepuluh kali uji coba atau *running*.

Tabel 8.1 Lokasi 1 – Makassar GA

Kapal	1	1	1	2	2	2	3	3	3	3	4	4	4	4	5	5	5	6	6	6
Tujuan	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Slot	2	3	4	1	7	8	1	2	4	5	2	3	4	5	2	3	5	1	2	3
Kapasitas	65 0	65 0	65 0	29 6	29 6	29 6	50 4	50 4	50 4	50 4	51 8	51 8	51 8	51 8	50 4	50 4	50 4	40 6	40 6	40 6
Running ke	Muatan																			
1	0	0	0	0	0	0	0	0	0	0	0	96	0	0	0	0	0	0	0	0
2	0	0	0	0	26 1	0	25 1	47 7	22 2	0	0	0	0	0	0	44 5	0	0	0	0

A-2

Kapal	1	1	1	2	2	2	3	3	3	3	4	4	4	4	5	5	5	6	6	6
Tujuan	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Slot	2	3	4	1	7	8	1	2	4	5	2	3	4	5	2	3	5	1	2	3
Kapasitas	650	650	650	296	296	296	504	504	504	504	518	518	518	518	504	504	504	406	406	406
Running ke	Muatan																			
3	0	0	0	268	0	0	0	0	0	0	0	0	172	124	0	0	3	0	0	0
4	0	0	0	0	0	0	0	0	0	29	227	0	0	382	0	0	0	261	0	306
5	0	0	0	0	0	0	0	0	0	0	0	83	0	0	48	0	47	0	0	0
6	149	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
7	0	0	0	0	0	224	0	0	0	0	0	0	0	0	0	0	0	0	246	211
8	0	0	487	0	0	0	0	0	0	0	0	496	0	0	0	0	0	0	0	319

Kapal	1	1	1	2	2	2	3	3	3	3	4	4	4	4	5	5	5	6	6	6
Tujuan	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Slot	2	3	4	1	7	8	1	2	4	5	2	3	4	5	2	3	5	1	2	3
Kapasitas	65 0	65 0	65 0	29 6	29 6	29 6	50 4	50 4	50 4	50 4	51 8	51 8	51 8	51 8	50 4	50 4	50 4	40 6	40 6	40 6
Running ke	Muatan																			
9	37 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	64 8	0	0	0	0	0	47 7	0	0	0	0	0	0	0	0	0		6	6

Kapal	10	10	10	10	11	11	11	11	11	11	11	12	12	12	12	12	12
Tujuan	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Slot	4	5	7	8	1	3	4	5	8	10	11	1	5	6	7	9	10
Kapasitas	283	283	283	283	211	211	211	211	211	211	211	219	219	219	219	219	219
Running ke	Muatan																
1	0	0	0	0	0	0	0	0	0	0	0	0	10	35	0	0	0
2	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0
3	0	0	277	227	0	39	0	0	0	0	0	0	0	0	0	213	193
4	165	100	0	0	0	0	197	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	79	0	0	146	0	0	0	0	0	0	0	0
6	0	269	0	283	0	0	0	186	0	0	0	0	0	0	0	0	0
7	0	0	0	0	129	0	0	0	0	0	0	0	0	0	173	0	0
8	0	0	0	197	0	0	0	0	0	0	0	175	0	0	0	0	0
9	0	0	0	0	206	0	0	93	201	0	191	0	0	0	0	0	0
10	258	0	0	0	0	0	0	176	0	0	0	0	0	0	0	0	0

Tabel 8.2 Lokasi 2 – Bitung GA

[illegible]

Kapal	8	8	9	9	9	10	10	11	11	11	11	11	12	12	12	12
Tujuan	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Slot	1	2	1	3	4	4	5	2	4	5	6	7	1	3	5	6
Kapasitas	538	538	330	330	330	283	283	211	211	211	211	211	219	219	219	219
Running ke		Muatan														
1	0	0	0	0	0	0	239	0	0	0	0	182	0	0	0	0
2	0	0	0	0	0	146	0	0	0	0	0	0	135	0	0	171
3	0	0	0	67	0	0	0	0	0	0	0	0	0	0	0	0
4	0	507	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	395	0	0	0	200	0	0	0	182	0	0	0	0	21	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	166	0	0	0	71	0	0
8	0	0	269	0	0	0	0	0	0	0	84	0	0	0	107	0
9	0	0	225	0	0	0	0	60	193	0	0	0	0	0	0	0
10	0	0	204	0	0	0	0	0	0	0	0	0	0	0	88	0

Kapal	9	10	10	10	10	11	11	11	11	12	12	12
Tujuan	3	3	3	3	3	3	3	3	3	3	3	3
Slot	3	1	2	3	4	1	3	4	5	1	2	3
Kapasitas	330	283	283	283	283	211	211	211	211	219	219	219
Running ke	Muatan											
1	81	0	0	0	0	0	0	0	149	0	0	0
2	0	0	0	0	0	0	0	0	130	0	0	0
3	0	0	273	0	175	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	204	0	0	0	0
5	0	147	0	0	0	203	0	0	0	0	0	198
6	0	0	0	94	0	0	0	0	0	0	0	0
7	314	0	0	0	0	0	0	0	0	0	182	0
8	0	0	276	0	172	183	0	0	0	0	0	0
9	0	0	266	0	234	0	0	0	0	0	0	0
10	311	0	0	0	0	0	103	0	0	104	0	154

Tabel 8.4 Lokasi 4 – Banjarmasin GA

Kapal	1	1	2	2	3	4	5	6	7	7
Tujuan	4	4	4	4	4	4	4	4	4	4
Slot	1	2	1	3	1	1	1	1	1	2
Kapasitas	650	650	296	296	504	518	504	406	538	538
Running ke	Muatan									
1	0	0	0	0	0	466	0	338	0	0
2	0	0	0	0	0	0	0	0	0	465
3	0	0	0	0	0	0	0	355	0	0
4	424	0	0	0	0	0	0	0	0	0
5	0	0	0	245	0	0	0	382	0	0
6	0	0	206	0	0	0	0	0	0	0
7	0	0	0	228	0	0	260	0	0	0
8	0	116	0	0	0	0	445	0	0	0
9	0	0	0	0	0	319	0	0	334	0
10	0	0	0	0	383	0	0	0	457	0

Kapal	8	8	9	9	10	10	10	11	11	11	12
Tujuan	4	4	4	4	4	4	4	4	4	4	4
Slot	1	2	2	3	1	2	3	1	2	3	3
Kapasitas	538	538	330	330	283	283	283	211	211	211	219
Running ke	Muatan										
1	0	0	0	0	0	0	0	0	36	0	0
2	0	0	0	0	0	0	214	0	0	0	161
3	485	0	0	0	0	0	0	0	0	0	0
4	0	251	0	0	0	0	0	165	0	0	0
5	0	0	0	0	0	213	0	0	0	0	0
6	205	0	0	280	0	0	0	0	0	149	0
7	0	0	0	149	0	0	203	0	0	0	0
8	0	0	0	0	279	0	0	0	0	0	0
9	0	0	187	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0

LAMPIRAN B

Lampiran ini adalah hasil analisis kromosom yang berisi muatan dari masing-masing kapal yang harus dibawa menuju tujuan satu yaitu Makassar, tujuan dua Bitung, tujuan tiga Ambon dan tujuan empat Banjarmasin. Kromosom-kromosom ini di hasilkan dari proses uji coba kombinasi Algoritma Genetika dan *tabu search* dengan menggunakan parameter N 450, Pc 0.8, dan Pm 0,1 yang dilakukan sebanyak sepuluh kali uji coba atau *running*.

Tabel 8.5 Lokasi 1 – Makassar GA-TS

Kapal	1	6	6	6	7	7	7	7	8	8	8	8	9
Tujuan	1	1	1	1	1	1	1	1	1	1	1	1	1
Slot	4	4	5	6	1	2	3	5	2	3	4	5	1
Kapasitas	650	406	406	406	538	538	538	538	538	538	538	538	330
Running ke	Muatan												
1	569	0	0	0	0	0	525	0	0	0	0	0	304
2	0	0	0	383	0	0	0	0	0	0	0	503	0
3	0	0	0	0	535	0	0	0	0	0	0	0	0
4	557	0	0	358	0	0	0	0	0	499	0	0	0

B-2

Kapal	1	6	6	6	7	7	7	7	8	8	8	8	9
Tujuan	1	1	1	1	1	1	1	1	1	1	1	1	1
Slot	4	4	5	6	1	2	3	5	2	3	4	5	1
Kapasitas	650	406	406	406	538	538	538	538	538	538	538	538	330
Running ke	Muatan												
5	0	371	0	377	0	0	0	0	0	0	420	460	0
6	0	0	0	405	0	538	0	0	0	0	488	462	0
7	540	0	332	382	0	0	0	0	0	0	0	442	0
8	580	0	0	0	0	0	0	0	418	0	417	437	0
9	0	0	356	359	0	0	0	0	0	0	417	466	0
10	0	0	342	0	0	0	0	451	0	486	0	467	0

Kapal	9	9	10	10	10	10	10	10	10	10	12	12	12
Tujuan	1	1	1	1	1	1	1	1	1	1	1	1	1
Slot	6	7	1	2	3	4	5	6	7	8	5	9	10
Kapasitas	330	330	283	283	283	283	283	283	283	283	219	219	219
Running ke	Muatan												
1	0	0	0	0	0	237	0	0	0	247	140	0	158
2	0	0	0	273	0	0	208	279	256	278	0	0	0
3	0	0	188	0	194	171	184	200	203	193	0	155	157
4	0	305	0	0	0	0	0	0	212	249	0	0	0
5	0	0	0	0	0	0	0	179	187	186	0	0	0
6	287	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	238	246	0	0	0
8	0	328	0	0	0	0	0	0	0	0	0	0	0
9	0	322	0	0	0	0	0	260	0	0	0	0	0
10	0	0	0	0	0	0	0	0	212	222	0	0	0

Tabel 8.6 Lokasi 2 – Bitung GA-TS

Kapal	1	1	6	6	6	6	7	8	8	9	10	10	10	10
Tujuan	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Slot	1	2	1	2	3	4	1	2	3	4	1	3	4	5
Kapasitas	650	650	406	406	406	406	538	538	538	330	283	283	283	283
Running ke	Muatan													
1	0	0	0	0	347	0	0	516	417	0	0	0	0	0
2	0	579	389	0	0	0	0	0	0	312	0	0	0	0
3	0	0	0	0	0	0	0	0	457	287	0	253	0	283
4	0	557	0	0	0	0	471	0	0	0	252	0	0	0
5	0	567	0	0	0	0	440	0	0	0	0	0	0	273
6	0	0	0	350	0	364	0	0	0	285	0	0	0	281
7	0	0	0	0	0	0	0	0	479	0	279	0	277	245
8	0	573	0	0	0	0	0	0	0	0	0	226	215	266
9	0	0	0	0	0	0	0	531	0	297	0	0	175	277
10	0	626	0	0	0	378	0	0	0	0	0	0	0	276

Tabel 8.7 Lokasi 3 –Ambon GA-TS

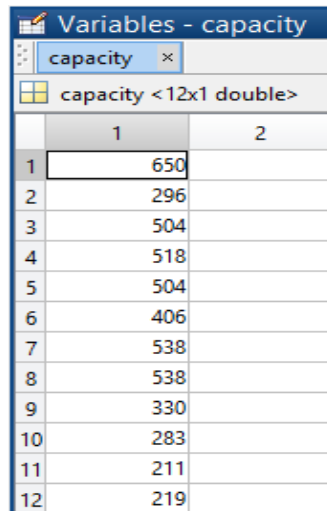
Kapal	1	6	6	7	8	8	9	9	10	10	12	12
Tujuan	3	3	3	3	3	3	3	3	3	3	3	3
Slot	2	2	3	1	1	2	1	3	3	4	1	4
Kapasitas	650	406	406	538	538	538	330	330	283	283	219	219
Running ke	Muatan											
1	597	0	0	0	0	0	303	0	0	0	0	0
2	562	0	0	0	0	0	0	0	0	181	157	0
3	0	0	0	0	0	500	0	0	226	174	0	0
4	0	0	0	472	0	428	0	0	0	0	0	0
5	0	0	0	0	0	0	0	314	172	280	0	134
6	557	0	343	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	467	0	0	209	224	0	0
8	0	369	0	0	0	531	0	0	0	0	0	0
9	0	0	0	0	440	460	0	0	0	0	0	0
10	0	0	0	515	0	0	0	0	189	196	0	0

Tabel 8.8 Lokasi 4 – Banjarmasin GA-TS

Kapal	6	7	8	8	9	10	10	11	12
Tujuan	4	4	4	4	4	4	4	4	4
Slot	3	2	1	2	3	2	3	4	3
Kapasitas	406	538	538	538	330	283	283	211	219
Running ke	Muatan								
1	392	0	0	448	0	0	0	0	0
2	0	0	0	410	0	196	234	0	0
3	341	0	0	0	0	274	225	0	0
4	0	0	507	0	0	0	226	107	0
5	337	0	503	0	0	0	0	0	0
6	360	480	0	0	0	0	0	0	0
7	369	0	0	471	0	0	0	0	0
8	401	0	0	0	306	0	0	0	133
9	0	0	415	425	0	0	0	0	0
10	388	0	0	452	0	0	0	0	0

LAMPIRAN C

Pada lampiran ini menjelaskan data perusahaan yang sudah di olah dan menjadi input dalam MATLAB. Terdapat tiga data input yaitu data kapasitas kapal, data harga sewa kontainer, dan data permintaan.

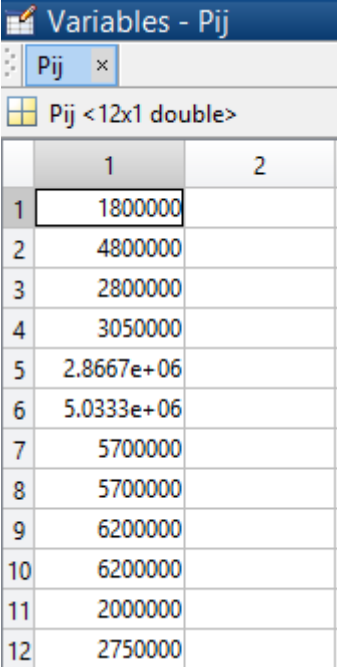


The image shows a MATLAB 'Variables' window titled 'Variables - capacity'. It displays a variable named 'capacity' of type 'double' with dimensions '12x1'. Below this, a table shows the values for each row of the array. The table has two columns: an index column (1 to 12) and a value column (650, 296, 504, 518, 504, 406, 538, 538, 330, 283, 211, 219).

	1	2
1	650	
2	296	
3	504	
4	518	
5	504	
6	406	
7	538	
8	538	
9	330	
10	283	
11	211	
12	219	

Gambar 8.1 Data Kapasitas Kapal

C-2



Variables - Pij

Pij ×

Pij <12x1 double>

	1	2
1	1800000	
2	4800000	
3	2800000	
4	3050000	
5	2.8667e+06	
6	5.0333e+06	
7	5700000	
8	5700000	
9	6200000	
10	6200000	
11	2000000	
12	2750000	

Gambar 8.2 Data Harga Sewa Kontainer

Variables - demand					
demand ×					
demand <1x4 double>					
	1	2	3	4	
1	2180	1280	900	840	

Gambar 8.3 Data Permintaan

LAMPIRAN D

Gambar 8.4 adalah hasil *running* GA untuk menampilkan gen dengan muatan yang paling sedikit pada “lowcap” dan muatan yang hampir penuh pada “fill_ship”.

```
lowcap =  
    115    121    178    191|  
  
fill_ship =  
Columns 1 through 15  
     7     10     23     28     36     37     61     82    100    109    120    141    142    154    167  
  
Column 16  
    179
```

Gambar 810.4 Hasil Gen Maksimal dan Minimal

LAMPIRAN E

Gambar 8.5 adalah lampiran hasil *running* pada MATLAB untuk proses GA

Command Window															
bestX1 =															
Columns 1 through 15															
1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	3	4	1	2	3	4	5	6	7	8	1	2	3	
0	0	0	0	0	0	216	0	0	272	0	0	0	0	0	
Columns 16 through 30															
3	3	4	4	4	4	4	5	5	5	5	5	6	6	6	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	
0	0	0	0	0	0	0	459	0	0	0	0	313	0	0	
Columns 31 through 45															
6	6	6	7	7	7	7	7	8	8	8	8	8	9	9	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	5	6	1	2	3	4	5	1	2	3	4	5	1	2	
0	0	0	0	0	278	461	0	0	0	0	0	0	0	0	
Columns 46 through 60															
9	9	9	9	9	10	10	10	10	10	10	10	10	11	11	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	4	5	6	7	1	2	3	4	5	6	7	8	1	2	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Gambar 8.5 Hasil GA Gen 1-60

E-2

Command Window														
Columns 61 through 75														
11	11	11	11	11	11	11	11	11	11	12	12	12	12	12
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	4	5	6	7	8	9	10	11	1	2	3	4	5	6
181	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Columns 76 through 90														
12	12	12	12	1	1	2	2	2	2	2	3	3	3	4
1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
7	8	9	10	1	2	1	2	3	4	5	1	2	3	1
0	0	0	0	0	0	283	0	0	0	0	0	0	0	0
Columns 91 through 105														
4	4	5	5	5	6	6	6	6	7	7	7	8	8	8
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	3	1	2	3	1	2	3	4	1	2	3	1	2	3
0	0	0	0	0	0	0	0	0	510	0	0	0	0	0
Columns 106 through 120														
9	9	9	9	10	10	10	10	10	11	11	11	11	11	11
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	2	3	4	1	2	3	4	5	1	2	3	4	5	6
0	0	0	249	0	0	0	0	0	47	0	0	0	0	170
Columns 121 through 135														
11	12	12	12	12	12	12	1	1	2	2	2	2	3	3
2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
7	1	2	3	4	5	6	1	2	1	2	3	4	1	2
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0

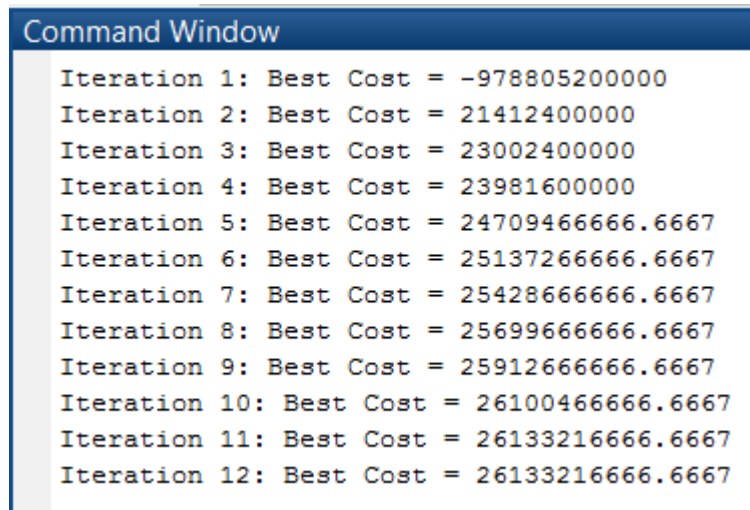
Gambar 8.6 Hasil GA Gen 61-135

Command Window														
Columns 136 through 150														
4	4	5	5	6	6	6	7	7	8	8	9	9	9	10
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
1	2	1	2	1	2	3	1	2	1	2	1	2	3	1
0	0	0	0	0	388	370	0	0	0	0	0	0	0	0
Columns 151 through 165														
10	10	10	11	11	11	11	11	12	12	12	12	12	1	1
3	3	3	3	3	3	3	3	3	3	3	3	3	4	4
2	3	4	1	2	3	4	5	1	2	3	4	5	1	2
0	0	0	142	0	0	0	0	0	0	0	0	0	0	0
Columns 166 through 180														
2	2	2	3	3	4	4	5	5	6	6	6	7	7	8
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1	2	3	1	2	1	2	1	2	1	2	3	1	2	1
0	159	0	0	0	0	0	0	0	0	0	0	248	377	0
Columns 181 through 195														
8	9	9	9	10	10	10	11	11	11	11	12	12	12	12
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
2	1	2	3	1	2	3	1	2	3	4	1	2	3	4
0	0	0	0	0	0	0	0	0	0	56	0	0	0	0
bestProfit =														
1.6091e+10														

Gambar 8.7 Hasil GA Gen 136 - 195

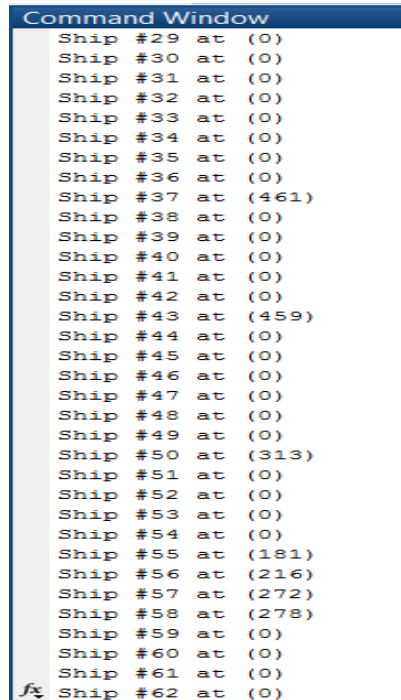
LAMPIRAN F

Gambar 8.8 adalah lampiran hasil *running* pada MATLAB untuk proses GA-TS.

A screenshot of a MATLAB Command Window. The title bar is dark blue with the text 'Command Window' in white. The window contains a list of 12 iterations, each showing the 'Best Cost' value. The values start at -978805200000 for iteration 1 and increase to 26133216666.6667 for iteration 12.

```
Iteration 1: Best Cost = -978805200000
Iteration 2: Best Cost = 21412400000
Iteration 3: Best Cost = 23002400000
Iteration 4: Best Cost = 23981600000
Iteration 5: Best Cost = 24709466666.6667
Iteration 6: Best Cost = 25137266666.6667
Iteration 7: Best Cost = 25428666666.6667
Iteration 8: Best Cost = 25699666666.6667
Iteration 9: Best Cost = 25912666666.6667
Iteration 10: Best Cost = 26100466666.6667
Iteration 11: Best Cost = 26133216666.6667
Iteration 12: Best Cost = 26133216666.6667
```

Gambar 8.8 Hasil Profit TS



The image shows a screenshot of a 'Command Window' with a list of ship IDs and their corresponding values in parentheses. The window has a blue title bar and a white background. The text is displayed in a monospaced font. The list starts with 'Ship #29 at (0)' and ends with 'Ship #62 at (0)'. The values in parentheses are: (0), (0), (0), (0), (0), (0), (0), (0), (461), (0), (0), (0), (459), (0), (0), (0), (0), (0), (0), (313), (0), (0), (0), (181), (216), (272), (278), (0), (0), (0), and (0). A small 'fx' icon is visible in the bottom left corner of the window.

Command Window	
Ship #29	at (0)
Ship #30	at (0)
Ship #31	at (0)
Ship #32	at (0)
Ship #33	at (0)
Ship #34	at (0)
Ship #35	at (0)
Ship #36	at (0)
Ship #37	at (461)
Ship #38	at (0)
Ship #39	at (0)
Ship #40	at (0)
Ship #41	at (0)
Ship #42	at (0)
Ship #43	at (459)
Ship #44	at (0)
Ship #45	at (0)
Ship #46	at (0)
Ship #47	at (0)
Ship #48	at (0)
Ship #49	at (0)
Ship #50	at (313)
Ship #51	at (0)
Ship #52	at (0)
Ship #53	at (0)
Ship #54	at (0)
Ship #55	at (181)
Ship #56	at (216)
Ship #57	at (272)
Ship #58	at (278)
Ship #59	at (0)
Ship #60	at (0)
Ship #61	at (0)
Ship #62	at (0)

Gambar 8.9 Hasil Muatan TS